$ 120.000

Micro Computación

AÑO 1 Nº 6

BASE DE DATOS

Qué es una base de Datos.

Informe histórico de las cuentas a cobrar.

El crecimiento de las familias de los procesadores.

Aumentan las opciones del sistema.

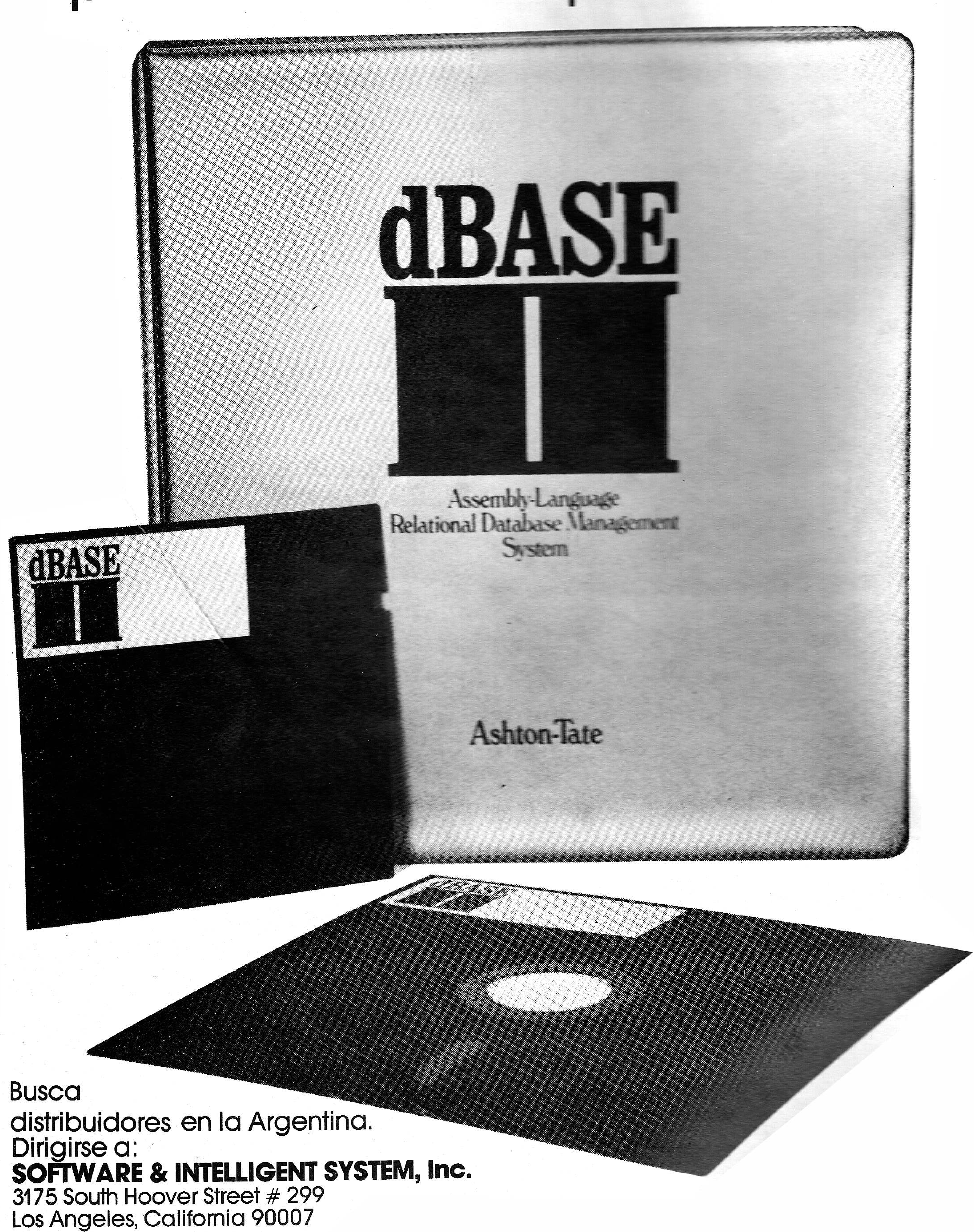
Soluciones mínimas a los dilemas lógicos.

En este número: PLANEOCALC programas comerciales

3 Cursos: BASIC - Comp. Digitales Electrónica Digital

VisiCalc

La más poderosa base de datos para su microcomputadora.





Director-Editor responsable
Antonio Cuevas

Dirección de Arte
Coordinación Editorial
Gustavo A. Otero

Jefe Publicidad:
Beatriz Abigador

Corresponsales:
Cristina Flores (California, EE.UU.)
Liliana Hembold (París)

Colombia
Alectronic Computer Limitada
Apartado postal 52100
Bogotá, Colombia
Sr. R. Rosoml Eckmard

Venezuela
Organización Playa Cía. Anónima
Apartado Postal 76904
Caracas, Venezuela
Sr. Luis Plaza

Mexico
Computrade
Mar Blanco 1743
Lomas del Country
Guadalajara Jalizco, México
Sr. Moisés Oedria

Dto. suscripciones:
Gustavo F. García

Fotocomposición:
Diseño

MICROCOMPUTACION es una revista mensual editada por FUTURART S.A. (en form.) Piedras 1184, Buenos Aires. Tel. 27-7447. Registro Nacional de la Propiedad Intelectual en trámite. Director-Editor responsable: Antonio Cuevas.
Queda hecho el depósito que indica la Ley 11.723 de Propiedad Intelectual. Las menciones de modelos, marcas y especificaciones se realizan con fines informativos y técnicos, sin cargo alguno para las empresas que los comercializan y/o los representan. Al ser informativa su misión, la revista no se responsabiliza por cualquier problema que pueda plantear la fabricación, el funcionamiento y/o la aplicación de los sistemas y los dispositivos descritos. La responsabilidad de los artículos firmados corresponde exclusivamente a sus autores.
Precio de este ejemplar: **$ 120.000** Precio de la suscripción en la República Argentina: **$ 1.320.000.-**
Distribuidor en Córdoba: INFORMATICA Producciones, Salta 191, 5000 Córdoba. Teléfono: 29006.
Impresión: **FA.VA.RO. S.A.I.C. y F.**

Micro Computación

AÑO 1 Nº 6

Editorial

La lucha es dura..., las adversidades muchas..., pero contra viento y marea seguimos creciendo. Fieles a nuestros propósitos y con una inquebrantable vocación de servir.

Alimentados por esa creciente cantidad de lectores que día a día nos exige y nos alienta, y que día a día nos compromete a seguir creciendo.

Hace seis números éramos una idea, una ilusión, una esperanza. Ya somos una realidad tangible, con más ideas, más ilusiones y más esperanzas que nunca y más ganas de seguir creciendo.

Hoy con el orgullo de sentir que el éxito corona nuestro esfuerzo, y con la humildad de saber que se lo debemos a nuestros lectores y a quienes confiaron en MICROCOMPUTACION desde la hora cero, nos complacemos en anunciar:

1.– Que nuestros ejemplares aumentaron su volumen en 14 páginas.

2.– Qué MICROCOMPUTACION a partir de este número se distribuye en México, Colombia y, en EE.UU., en los estados de California y Miami.

3.– Que editorial FUTURART S.A. tiene una nueva sede más grande y más cómoda para atender a nuestros lectores y anunciantes, en la calle Piedras 1184 de Capital Federal.

4.– Que el día 1º de Mayo se pondrá en funcionamiento nuestro centro de procesamiento de DATOS que estará al servicio del lector y de las empresas proveedoras cuya forma de uso se explicará en el Nro. 7 de MICROCOMPUTACION.

5.– Que el día 1º de julio habilitaremos un moderno laboratorio electrónico para analizar y testear los distintos equipamientos que ofrece el mercado.

y 6.– ...Que aunque la lucha sea dura..., las adversidades muchas, contra viento y marea... ...Seguimos creciendo.

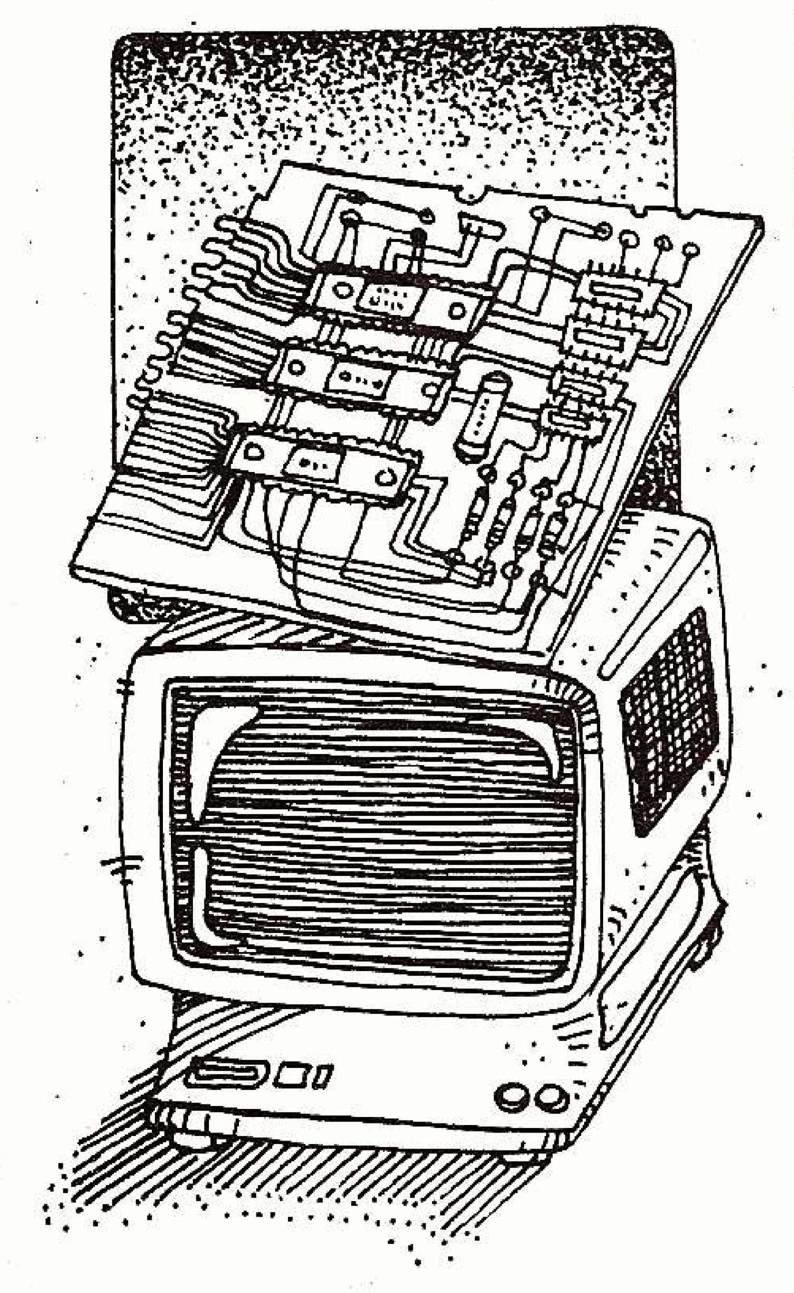
LA DIRECCION

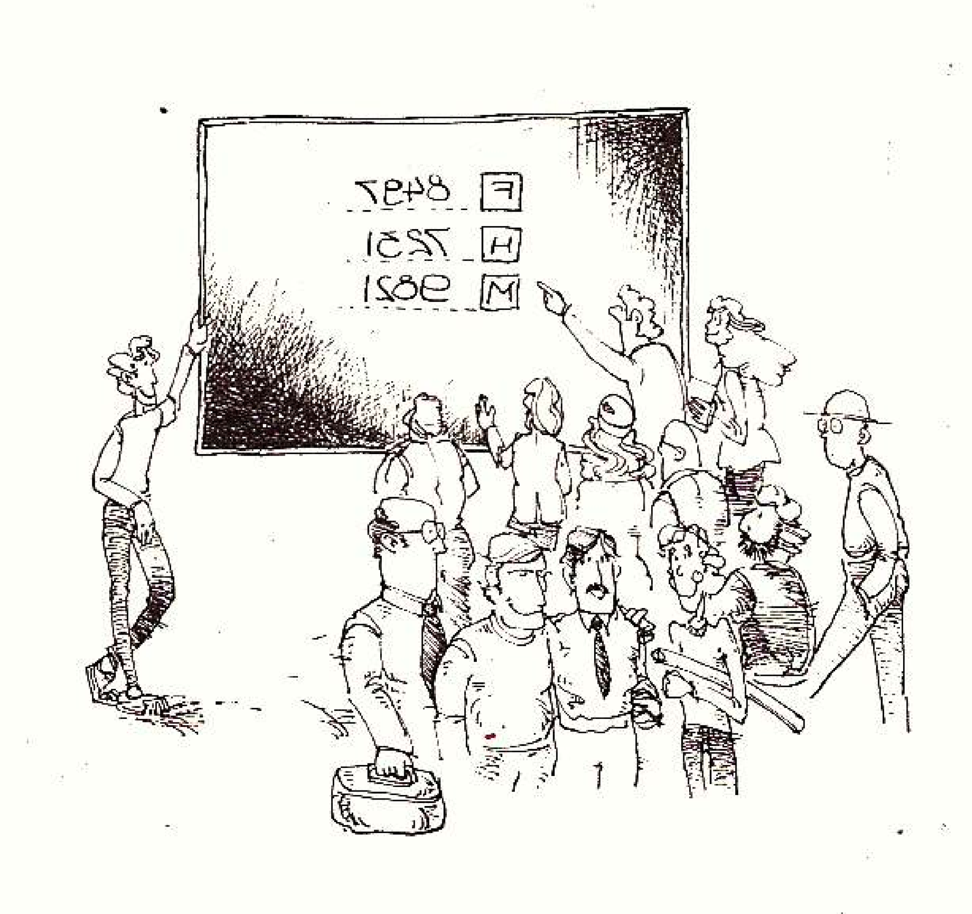
Micro Computación

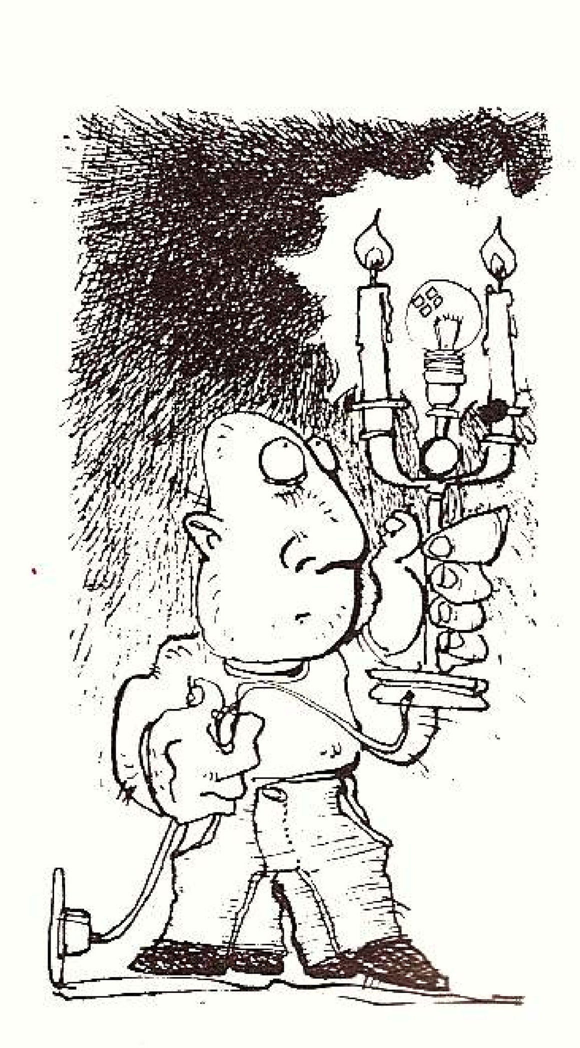
AÑO 1 Nº 6

Sumario









Aquí están los mejores accesorios magnéticos para su centro de cómputos.

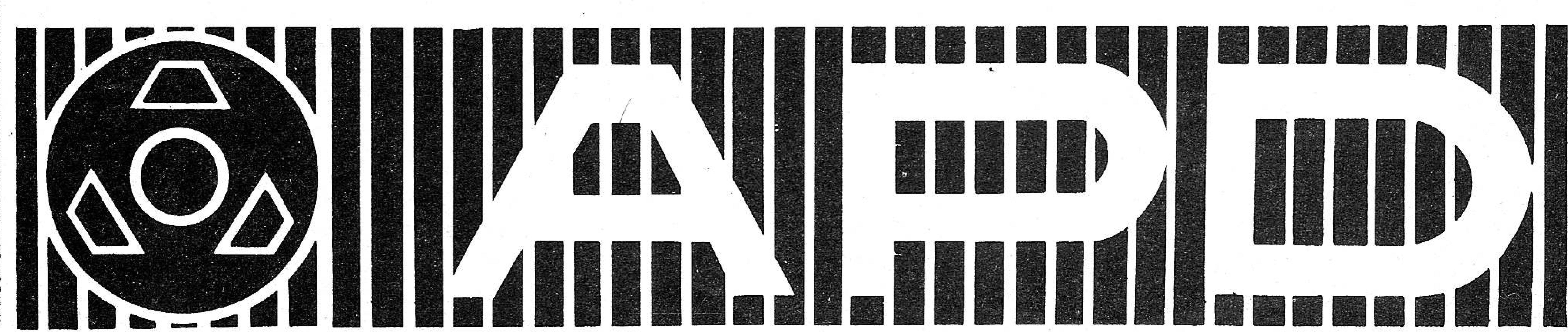
Si usted es conciente de la importancia de su centro de cómputos, no lo piense más, los mejores accesorios para su computadora los encontrará en nuestro local de venta. Le aseguramos calidad y seguridad en el procesamiento de su información.

Diskettes, Disk Pack, Disk Cartridge, Cassettes, Cintas Magnéticas, Cintas de Impresión, Formularios Contínuos, Carpetas de Archivo y Muebles.

ATHANA

UNICO DISTRIBUIDOR OFICIAL
AUTORIZADO EN LA
REPUBLICA ARGENTINA





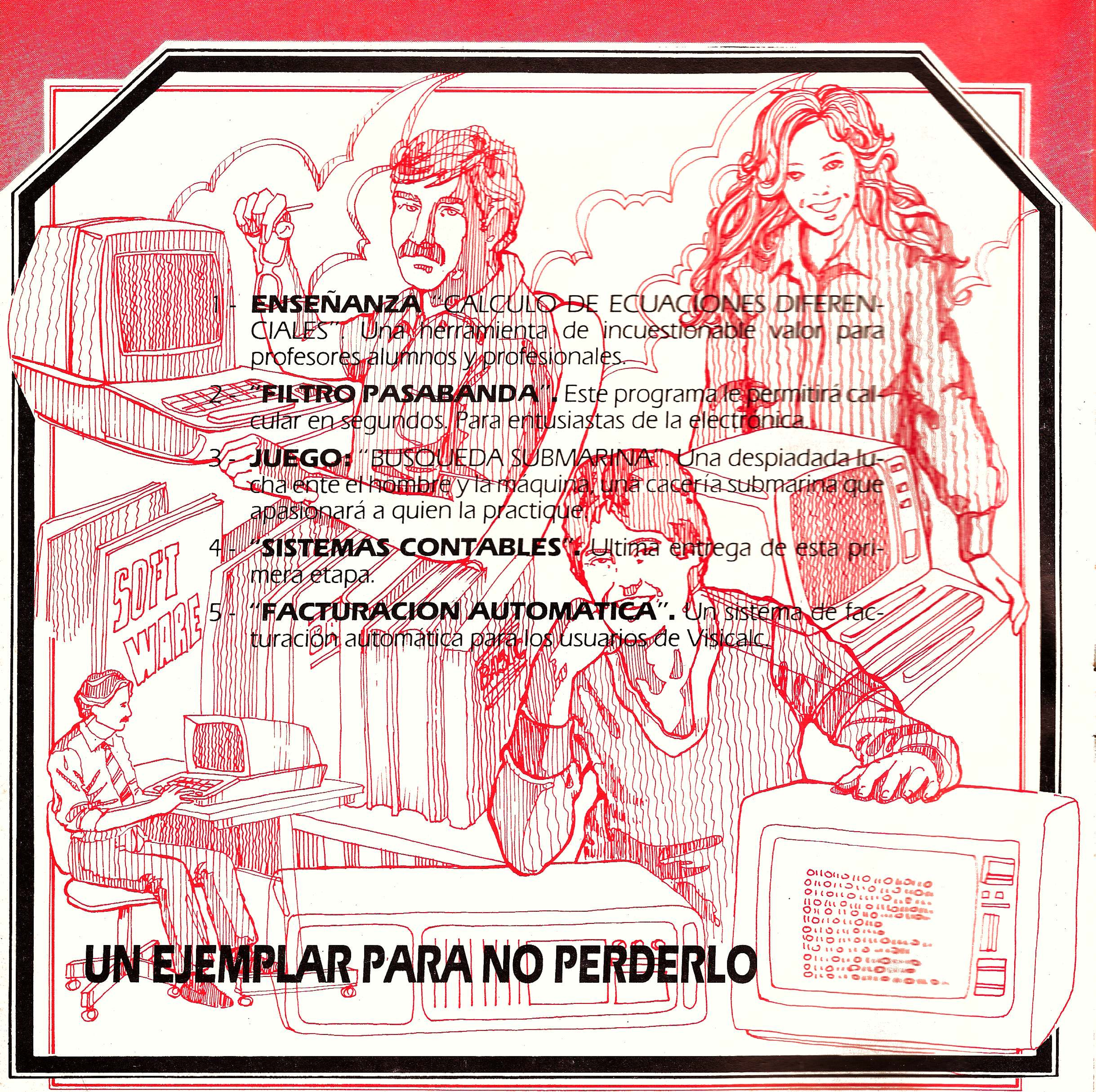
MARQUE CON UN CIRCULO EL 68 EN LA SERVICE CARD

ACCESORIOS PARA PROCESAMIENTO DE DATOS S.A.

Rodríguez Peña 330, Tel 46-4454/45-6533 (1020) Capital Federal

Próximo Número

5 PROGRAMAS LISTOS PARA USAR



1 - **ENSEÑANZA** "CALCULO DE ECUACIONES DIFERENCIALES". Una herramienta de incuestionable valor para profesores alumnos y profesionales.

2 - **"FILTRO PASABANDA".** Este programa le permitirá calcular en segundos. Para entusiastas de la electrónica.

3 - **JUEGO:** "BUSQUEDA SUBMARINA". Una despiadada lucha ente el hombre y la máquina, una cacería submarina que apasionará a quien la practique.

4 - **"SISTEMAS CONTABLES".** Ultima entrega de esta primera etapa.

5 - **"FACTURACION AUTOMATICA".** Un sistema de facturación automática para los usuarios de Visicalc.

UN EJEMPLAR PARA NO PERDERLO

QUE ES UNA BASE DE DATOS

Usted ya puede utilizar los programas de procesamiento de datos, hoja de desglose electrónicas (spread sheets), sistemas contables, ajedrez, analizadores de bioritmo, y muchos otros programas que operan en su nueva microcomputadora, pero hay pocas tareas comerciales importantes que permanecen fuera del alcance de estos programas. Por ejemplo, usted pudo seleccionar ciertos clientes de su sistema de cuentas a pagar con propósitos especiales de cobro. O puede desear clasificar a los clientes de acuerdo con el volumen de ventas y seleccionar clientes de bajo volumen de ventas para realizar para el envío de una promoción especial. Usted puede desear un informe impreso que indique los items de inventario con el precio más alto, el cliente que los compra, y las fechas de compra en los últimos seis y doce meses.

Un ejemplo más simple nos indicará las funciones básicas comunes a cada una de estas tareas. Si usted quiere guardar una lista de Navidad, usted podría comprar un programa de lista de envío por correo. Pero este programa no le ayudará a mantener los datos de talles de camisas, nombres de los niños, preferencias por ciertas marcas de licor, etc., etc. Para incluir tal información, usted puede decidir que ya es tiempo de aprender programación, y pasar cuatro horas por día escribiendo un programa BASIC que funcione, pero escasa e insuficientemente.

Almacenar este tipo de información en una lista de Navidad computarizada es un buen ejemplo de manejo de datos. Un archivo electrónico en un disco contiene información sobre sus amigos, particularmente sus nombres, direcciones, números telefónicos y preferencias personales. Un programa le permite manejar la información: agregar nuevos nombres, cambiar y borrar nombres ya existentes, e imprimir las identificaciones para las tarjetas.

Ahora supongamos que usted escribe otro programa BASIC para mantener un registro de su colección de discos, imprimiendo listas de albumes por artista y por tipo de música. Usted aprende rápidamente una de las técnicas más antiguas de programación: tomar un viejo programa y realizar en él un cambio pequeño para que realice algo nuevo. Usted puede necesitar solamente dos días para escribir el programa registro de discos por cuanto copió la estructura de listado de Navidad.

Afortunadamente, los expertos en computación hace ya mucho tiempo reconocieron la necesidad de solucionar estos problemas sin días de programación. Crearon **la base de datos**. Inicialmente diseñada para ayudar a programadores profesionales, la base de datos más tarde se implementó para los **usuarios** de la computación. En la actualidad, con una microcomputadora en casi todas las casas y oficinas, aún un no programador puede beneficiarse con la tecnología de la base de datos. Pero esto no excluye al programador. En realidad la base de datos incrementa la productividad de los programadores profesionales, permitiéndoles desarrollar sistemas para microcomputadoras que sean tan sofisticados como los que operan en grandes computadoras.

Una base de datos identifica un juego o compendio de información a la que la computadora puede acceder y operar. Es una herramienta para mejorar problemas como los dos descriptos anteriormente, problemas que involucran información almacenada en discos. No le puede ayudar con las amortizaciones de préstamos o con ecuaciones diferenciales; no podrá jugar juegos de computadora o invertir matrices, y no le permitirá eliminar todos los gabinetes de archivo de su oficina.

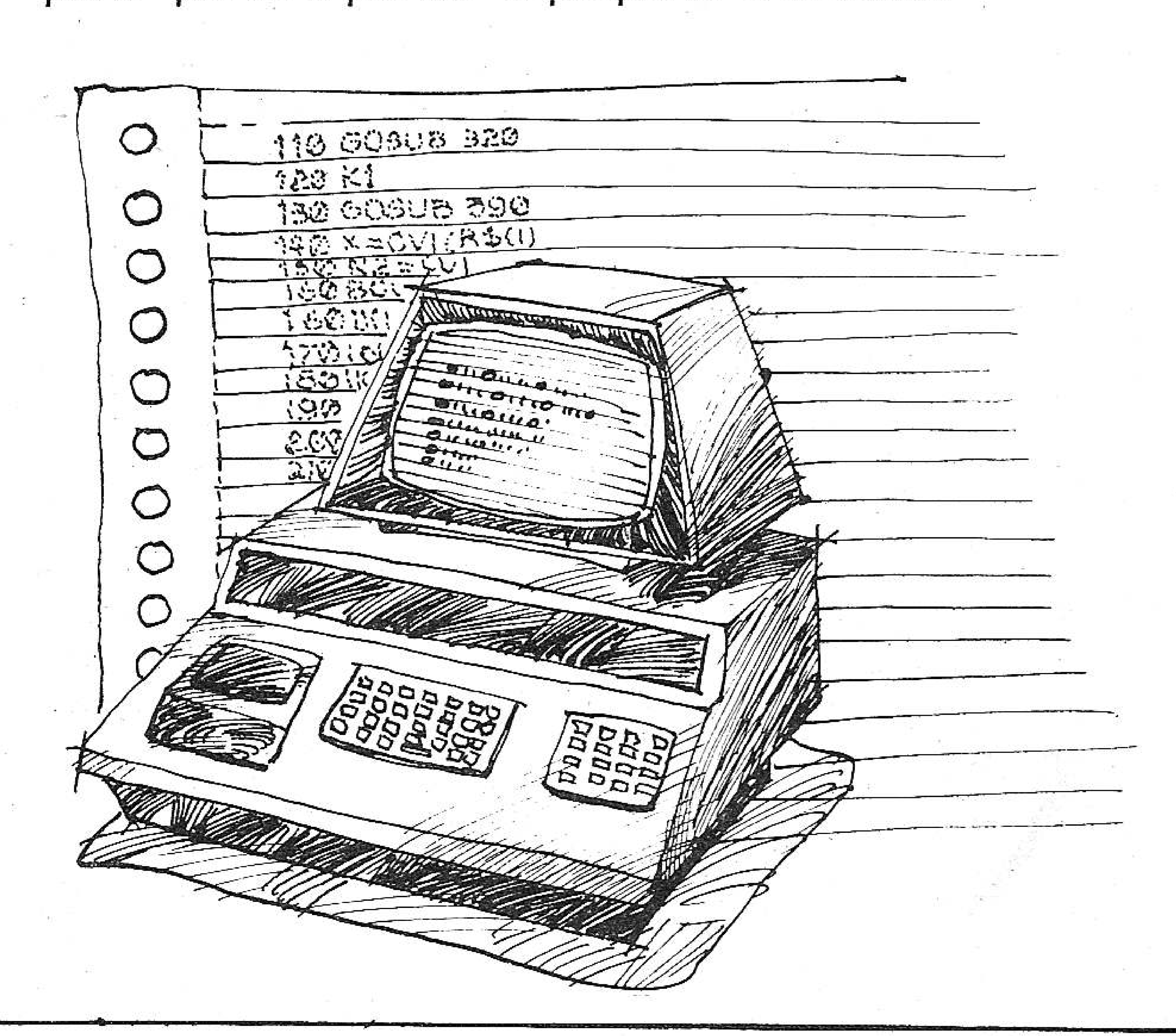
Sin embargo, una base de datos **le ayudará** a que parezcan triviales problemas tales como la lista de Navidad y el catálogo de discos (usted puede fijar uno de

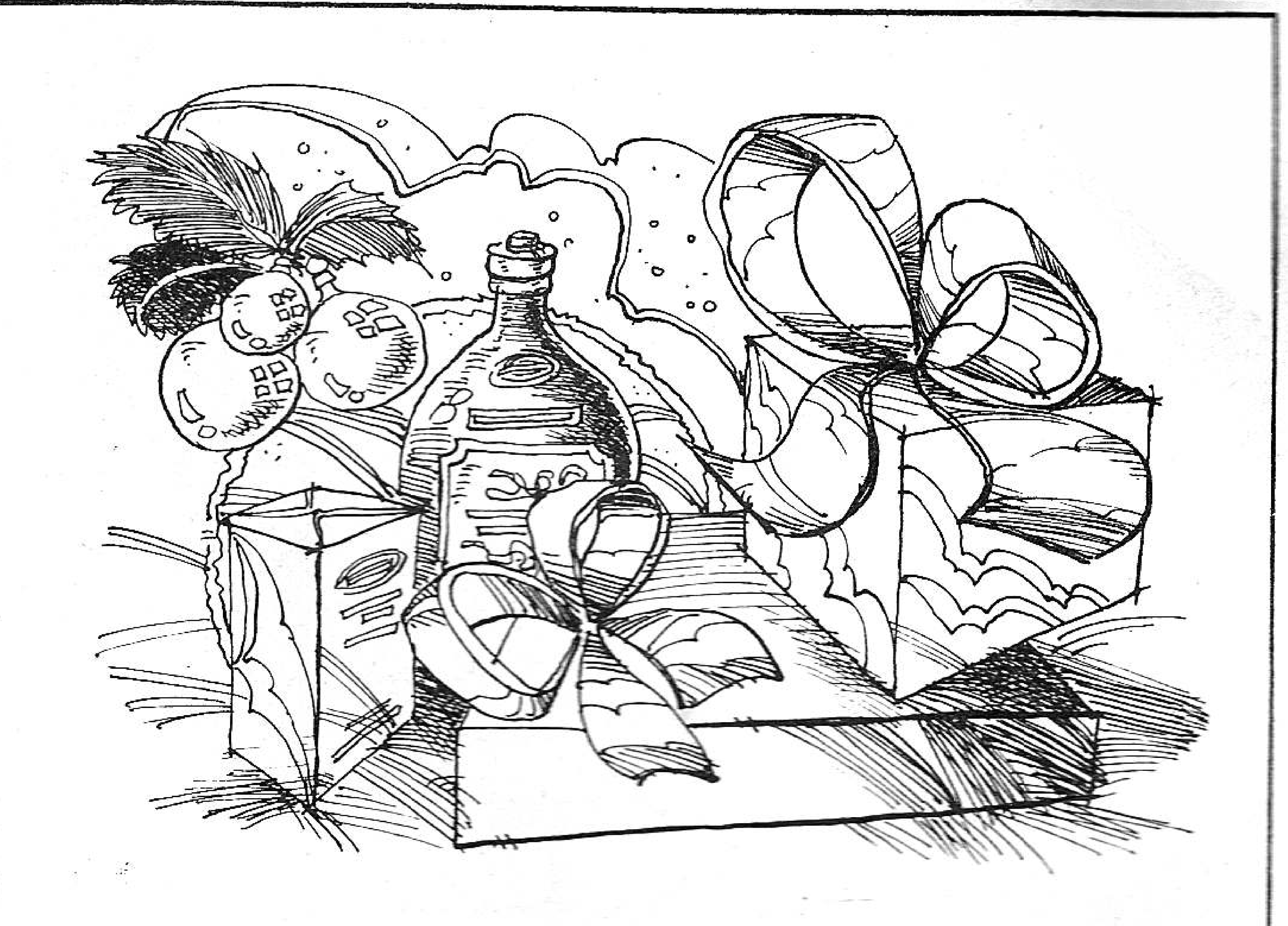
ellos en menos de una hora), y para que sean menos intimidantes las aplicaciones comerciales complejas, tales como el procesamiento de órdenes de ventas.

UNA BREVE HISTORIA DEL DESARROLLO DE LA BASE DE DATOS

El desarrollo del software para las microcomputadoras apareció aproximadamente quince años después del de los sistemas de las grandes computadoras y aproximadamente siete años después del de las minicomputadoras. Así como los departamentos de procesamiento de datos que utilizan programas COBOL standard, discos simples y archivos a cinta de grabación comienzan a darse cuenta el gasto del software inflexible, así también hoy en día, los propietarios de microcomputadoras que utilizan BASIC se encuentran con el mismo problema. Mientras que los fabricantes de computadoras rescataron a los usuarios de grandes sistemas de computadoras ofreciéndoles sistemas sofisticados a base de datos por 40.000 dólares o más, los editores de software ahora ofrecen paquetes de base de datos a los propietarios de microcomputadoras por 700 dólares. Tal como muchos comercios de procesamiento de datos aún luchan con el software desde la década del 60, en quince años algunos propietarios de microcomputadoras estarán usando el software del ayer.

Las computadoras centrales son frecuentemente apoyadas ("soportadas") por un gran personal de programadores profesionales. La Gerencia de una Compañía depende frecuentemente de su experiencia, y muchos comercios de procesamiento de datos desean preservar esa relación de dependencia. Reconociendo esta realidad política, las casas que comercializan software tradicionalmente no han ofrecido paquetes que permitieran a los usuarios finales de las computadoras dejar de lado a los programadores. Consecuentemente, los primeros sistemas de base de datos utilizaron los lenguajes existentes (por ejemplo, COBOL) y fueron complejos y difíciles de aprender. Sólo recientemente se han introducido los lenguajes con preguntas o interrogantes, programas que permiten a aquellas personas que no son programadores acceder a los datos pero que aún requieren personal de procesamiento de datos para "poner a punto" o preparar el sistema.





Una operación de base de datos es la recuperación de la información. Por ejemplo, el Departamento de Censos de los Estados Unidos vende cintas magnéticas que describen en un minuto los detalles del comercio exterior de los Estados Unidos. Luego de alimentar estos datos en la base de datos, un ejecutivo petrolero puede solicitar a la computadora que liste todos los embarques de petróleo al puerto de Los Angeles en orden decreciente de volumen y por país de origen. Utilizando las viejas técnicas de programación, el programador debería haber tenido que anticipar cada pregunta. Cada vez que el ejecutivo deseaba nueva información de los datos del comercio exterior, el programador debería haber tenido que escribir un nuevo programa. En el momento en que se hubiera terminado el programa, el ejecutivo habría olvidado por qué o para qué lo quería. La base de datos tambizn se utilizó en programas de aplicación tales como inventario, facturación, reservaciones de líneas aéreas y contabilidad. Los beneficios principales en la utilización de la base de datos en lugar de la programación convencional son los costos y el tiempo que se ahorran en la preparación o "puesta a punto" de los sistemas y los posteriores cambios a los mismos. Dado que los departamentos de procesamiento de datos utilizan la mayoría de su tiempo "manteniendo" viejos sistemas, cualquier cosa que reduzca semejante carga es bienvenida.

DEFINIENDO LA BASE DE DATOS

Puede surgir la pregunta de como una lista de tarjetas de Navidad de una microcomputadora pueda tener algo en común con un sistema de gestión de información, en una gran computadora. La respuesta es simple. En ambos casos una persona solicita y recibe información de la computadora. ¿Qué pasa dentro de la computadora para que esto sea posible?

Imagine, por ejemplo, que la computadora es una sala pequeña donde un duende pequeño tiene su oficina. En uno de los extremos de la oficina tiene un gran gabinete de archivo que contiene cada bit de información que alguien le ha suministrado. El gabinete de información tiene una indicación que dice "Base de Datos", y el duendecito se ha otorgado el título de "Gerente de la Base de Datos". Su trabajo es simplemente buscar cosas cuando la gente se lo solicita. Si us-

ted le pregunta a cual de sus amigos le gusta el whisky "Old Smuggler", obligatoriamente deberá buscar en el archivo de "Navidad" en el cajón inferior y tipear la lista de nombres.

Ahora imagine que usted va a trabajar el lunes a la mañana y le solicita a su contador una lista de quién le debe qué. El contador le solicita al gerente de base datos de la compañía que revea un mayor de cuentas a cobrar y que produzca la información requerida. El gerente de base de datos de la compañía comienza a trabajar, buscando en cada carpeta de cada comprador o cliente para determinar la cantidad que debe y la fecha de vencimiento. Ocasionalmente, una factura o un memorandum de crédito no tienen importancia, pero el contador es solamente una llamada telefónica para preguntas específicas. Por último, el gerente de base de datos termina el informe y se lo envía al contador, quien se lo entrega a usted. Unos pocos minutos más tarde, el gerente de ventas llama al gerente de base de datos solicitándole una lista de clientes en Perú en orden creciente de ventas de un año hasta la fecha. El gerente de base de datos retorna a los mismos archivos y arma la información.

Obviamente, el duendecito y el gerente de base de datos de la compañía son colegas. El gerente de base de datos de la compañía es en realidad un programa de computadora llamado Sistema de Gestión de Base de Datos (DBMS).

El gabinete de archivo es en realidad un disco que contiene la base de datos.

Un DBMS separa la información de los programas y de la gente que la usa y permite muchas visiones totalmente diferentes de la misma información. Este enfoque se ilustra en la figura 1-1; contrasta dramáticamente con el método más antiguo ilustrado en la figura 1-2 en el cual cada programa se conecta a la información física. En el primer caso, los programas tienen una **visión lógica** de la información que es independiente de como la información se coloca **físicamente** en el disco. La información en el disco se puede cambiar o suplementar sin necesidad de un cambio de programa.

Recuerde que DBMS es el programa de computación que usted compra y que la base de datos es su información luego de que DBMS la haya almacenado en un disco. Puede ser que tenga que escribir también sus propios programas, o no, esto depende de cual DBMS es el que usted elige.

OBJETIVOS DE DBMS

Deben haber beneficios diferentes al de la elegancia, que sean los que justifiquen el gasto de la compra y preparación de un DBMS. Consideren los siguientes beneficios como objetivos para el examen y evaluación de cada paquete DBMS futuro.

PRESENTA UNA FORMA O MODELO PRECISO DE LA INFORMACION

Presumiblemente un DBMS no existiría si no presentara una forma o modelo preciso de la información de alguien. Si usted tiene una sola aplicación en mente para su DBMS, usted puede equiparar el istema a sus requerimientos, pero si obtiene un DBMS para sus para propósitos generales, usted tiene algo de trabajo que deberá realizar. El capítulo 2 le ayudará a comprender la fortaleza y la debilidad de algunas estructuras de información muy diferentes.

ORGANIZA LA INFORMACION CON SIMPLICIDAD

La estructura de la información en un DBMS debería ser lo más simple posible. Cuanto más simple es la estructura de la base de datos, más fácil es manejar la información.

SUMINISTRA RESPUESTAS OPORTUNAS A LAS PREGUNTAS

Un DBMS debe operar lo suficientemente rápido. Si usted exhibe información en la pantalla, una demora de dos segundos es aceptable. Si usted necesita un informe que involucre la búsqueda de toda la información en un archivo, usted debe ser el juez en cuanto a cuanto es el tiempo que debe tardar.

¿Cuánto es para usted un tiempo excesivo? Cinco minutos, diez minutos, quizás varias horas. La base de datos puede preparar o "poner a punto" de tal manera que las preguntas frecuentes son "baratas" y las preguntas menos frecuentes son "caras". Realizando una pequeña investigación y experimentación con información simulada, usted puede contestar estas preguntas antes de comprar un paquete particular DBMS.

REDUCE COSTOS

El almacenamiento en disco es cada vez más barato, y los programadores son cada vez más caros. La mayoría de los paquetes DBMS reduce la necesidad de cualquiera de ellos. Mientras que un DBMS reduce realmente la redundancia de informacón, usted puede darse cuenta realmente del espacio extra del disco por cuanto en los sistemas previos a la base de datos, la información redundante se almacenaba normalmente en archivos temporarios. Por ejemplo, para realizar una lista de los empleados en orden numérico y alfabético no debería haber mantenido dos archivos de empleados. En cambio debería haber clasificado el archivo numérico primario en un listado alfabético temporario para su impresión. A la inversa, en el medio ambiente DBMS, un archivo de empleados parece estar en dos órdenes diferentes de clasificación.

El ahorro real está en el tiempo de programación. Con un DBMS, las soluciones a los problemas simples toman horas en cambio de días. Las aplicaciones complejas requieren un programador en cambio de un grupo de ellos. Reduciendo el costo de un juego de programas comerciales de 6.000 a 3.000 dólares (incluyendo el costo de un DBMS) puede hacer posible para una compañía pequeña realizar una inversión en programación.

UTILIZA INFORMACION NO REDUNDANTE SOLAMENTE

La información redundante requiere poco espacio en disco pero tiene un gran potencial para los errores. Si usted registra un envío por diez unidades del producto X y usted inmediatamente disminuye las "unidades en stock" del producto en diez, todo está correcto a menos que se cancele el envío. En ese caso usted deberá

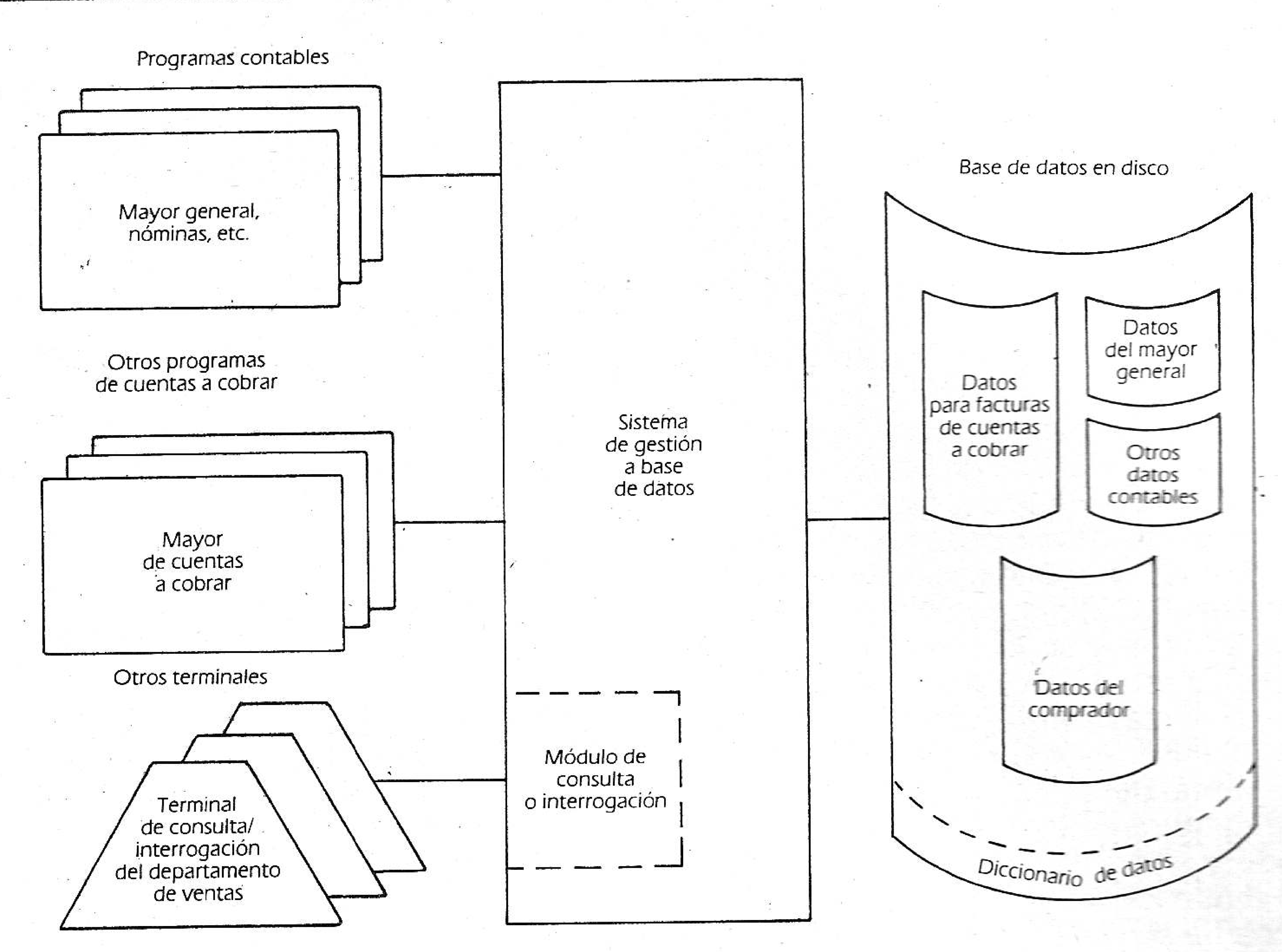


FIGURA 1-1: El medio ambiente de la base de datos.

agregar diez unidades nuevamente al inventario. En un buen DBMS, no existe ningún número de unidades en stock. En cambio, usted mira el balance del principio de mes, luego sume todos los envíos que tiene normalmente en archivo, y reste. Si se cancela un pedido, simplemente la cantidad del pedido no se resta.

PROTEGE LA INTEGRIDAD DE LA INFORMACION

Una base de datos frecuentemente se compone de un número de archivos, registros, items de información, e interconexiones. Si algo anda mal, por ejemplo, una falla de la energía, puede llegar a no poderse leer una parte de la base de datos. El DBMS se debería verificar a si mismo e inmediatamente hacer saber si encuentra algo que realmente "anda mal". El enfoque o acercamiento más común involucra la "apertura" y "cierre" de la base de datos. Por ejemplo, antes de que las transacciones comerciales terminen el Viernes, el operador de la computadora termina (cierra) el procesamiento de la base de datos y DBMS y el operador registran la hora y la fecha. Cuando la base de datos se reactiva (abre) el Lunes por la mañana, exhibe la información del cierre anterior. El operador puede verificar si la fecha y la hora concuerdan con las entradas en el libro de registros. Si la base de datos se abre dos veces en una fila sin un cierre de intervención, entonces DBMS por sí mismo sabe que hay un problema.

En un nivel más bajo, si la base de datos tiene lugar para los precios, no tendría sentido que el precio sea menor a cero, ni tampoco sería razonable que el precio fuera XYZ o cualquier otro valor no numérico. El DBMS debería estar siempre buscando u observando informacion no-compatible no permitiendo su entrada, y, en caso de que ésta se deslizara, alertar al usuario de su presencia.

En aquellos tiempos, una base de datos se conocía como **un sistema de gestión de información**. La idea era poner todos los datos de la compañía en una vasta pileta donde los ejecutivos pudieran "ir de pesca" a obtener cualquier información que desearan. Por supuesto, la información reflejaría instantáneamente cualquier cambio que resulte de los pagos de sueldos, facturación de ventas, etc. Pero este sistema no era adecuado por cuanto el software no era sofisticado en la medida necesaria y dado que los ejecutivos no conocían cuales eran las preguntas que debían realizar. Luego de fallar en la implementación de un sistema general de gestión de la información, los profesionales de la computación reconsideraron y aplicaron el concepto de la base de datos a modelos específicos de operaciones.

INTERCONECTA CON EL PASADO

Si usted no ha tenido nunca una computadora, no tiene que precocuparse por esto.

Sin embargo, si usted tiene un juego de programas e información para transferir a la base de datos, puede pasar que usted no desee volver a ingresar toda la información. Un DBMS estaría capacitado para aceptar la información de los archivos convencionales, y estaría capacitado para entregar la información en un formato de archivo para su utilización por programas existentes que no sean de base de datos.

Si usted posee un juego de programas convencionales de contrabilidad operando en su microcomputadora, usted probablemente tenga un archivo de vendedores, uno de compradores o clientes, un master de nóminas, y un archivo general contable del mayor. Usted puede colocar todos estos archivos en la base de datos y luego reacomodarlos como sea necesario para preguntas e informes que su viejo sistema nunca le sumi-



INDEPENDENCIA
financia su crecimiento
Hable con su proveedor,
plantéele sus necesidades
de equipamiento, de **financiación**,
hable con nosotros
INDEPENDENCIA
La Rioja 827 - Buenos Aires (1221)
Tel. 97-7652 / 4257 / 8280
MARQUE CON UN CIRCULO EL 75 EN LA SERVICE CARD

nistró. Hasta que usted realice la conversión total de su sistema a un sistema contable de base de datos en su totalidad, tendrá que conservar copias de los archivos a medida que el viejo sistema los vaya actualizando.

INTERCONECTA CON EL FUTURO

Sin un DBMS, aún los cambios más triviales pueden requerir volver a escribir docenas de programas. Una vez que se realiza esa re-escritura, aparecen nuevos errores y defectos, algunas veces los mismos causan detención del equipo y también se pierde información.

Los sistemas de gestión de base de datos pueden hacer que los cambios sean menos penosos. Frecuentemente usted puede recomponer o agregar elementos de información sin tocar ningún programa excepto aquéllos que utilizan esos nuevos elementos. La mayoría de las demandas en un sistema de trabajo son solicitudes de nuevos informes. Si la base de datos tiene un poderoso generador de informes, estas solicitudes se pueden realizar en horas.

PERMITE EL USO COMPARTIDO DE LA INFORMACION

El DBMS permitiría que se comparta la información. En una organización con una computadora de un solo terminal, un DBMS permitiría que diferente personal utilice la misma información. Quizás el personal de compras y el empleado de cuentas a pagar puedan compartir el mismo archivo del vendedor. Con un sistema de multiusuario, una persona de ventas puede imprimir una lista de los clientes más activos desde un terminal mientras que un empleado verifica el crédito del comprador de otro terminal.

OFRECE FLEXIBILIDAD DE LENGUAJE

Además del lenguaje de interrogación de alto nivel ya descripto anteriormente, existe un lenguaje de programación de bajo nivel que evita que los programadores digan "Nosotros no podemos hacer eso". El lenguaje de bajo nivel puede ser un lenguaje establecido, como ser COBOL o BASIC, con "enganches" dentro del DBMS o puede ser un lenguaje de marca registrada incluído en el paquete DBMS.

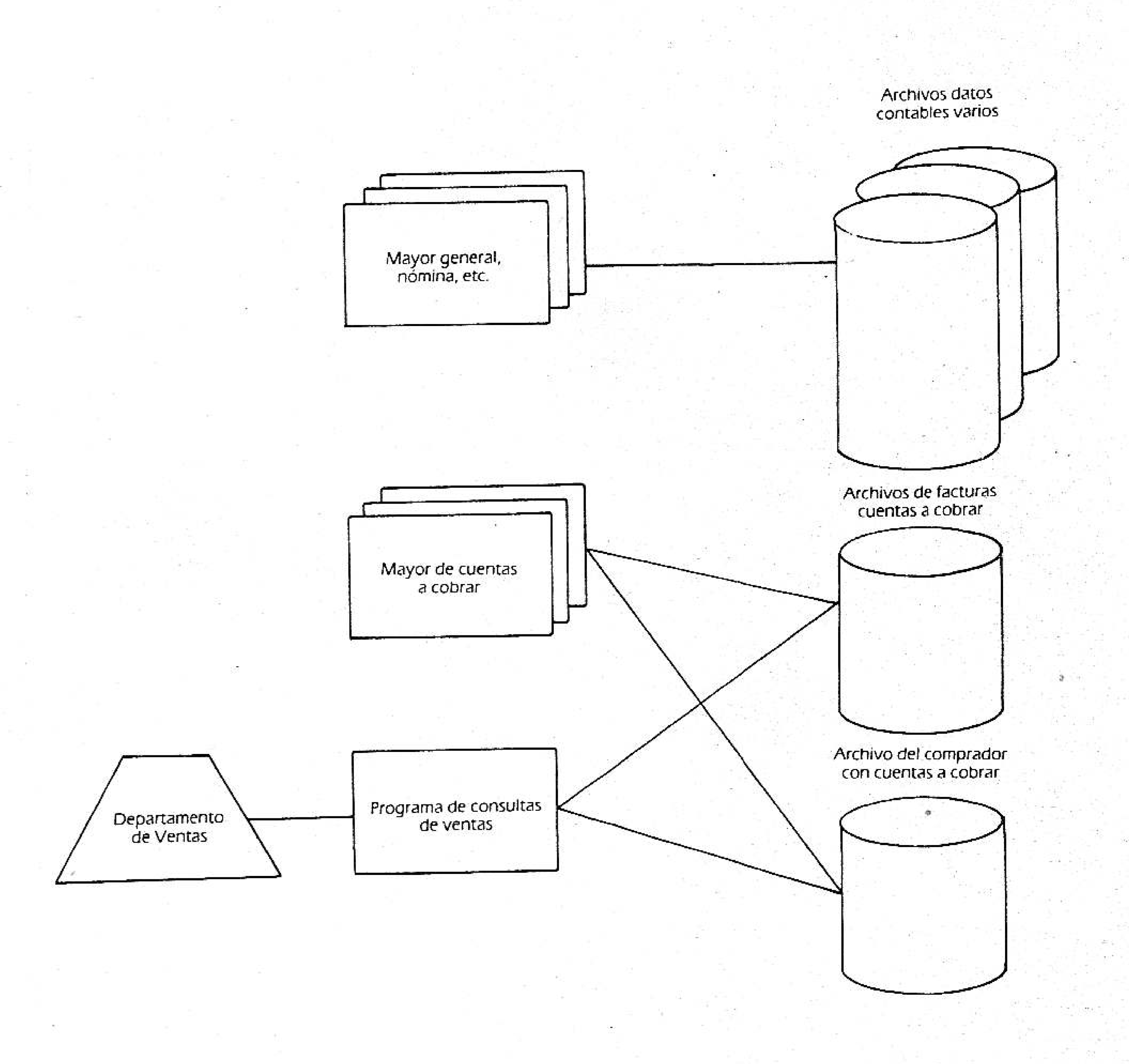
PRODUCE SU PROPIA DOCUMENTACION

Casi todos los programadores frecuentemente escriben sus sistemas que nadie más los entiende. Pueden estar motivados por la seguridad del trabajo o por ignorancia, pero el resultado es siempre inaceptable. Si el software produce automáticamente documentación, usted va a la cabeza del juego.

En todos los sistemas DBMS, el diccionario de información suministra un buen comienzo hacia la documentación producida por la computadora. Manteniendo un libro con copias impresas de estos diccionarios usted conocerá la estructura de su base de datos en todo momento. Requiere que el que haga un cambio realice un nuevo listado y que lo coloque en el libro.

LIMITACIONES DE LA MICROCOMPUTADORA

Como usted ha visto, algunos sistemas de gestión de base de datos de microcomputadoras cumplen con la mayoría de los objetivos fijados para los grandes sistemas de computadoras. Porque entonces usted podrá preguntarse por qué la gente paga millones por las grandes computadoras cuando en realidad pueden comprar una microcomputadora de 5000 dólares con un paquete de base de datos por 700 dólares. Una parte de las respuestas que se nos ocurren es **las redes**. Los sistemas de las grandes computadoras se pueden conectar a miles de terminales en todo el país. Pero una microcomputadora que opera bajo un CP/M está restringida a una computadora y un terminal.



FIGRUA 1-2: Programación orientada en archivo antes de la base de datos.

Aunque los fabricantes de microcomputadoras han puesto a la venta sistemas de computación en red y multiusuarios, muchos de estos sistemas simplemente comparten un disco rígido caro entre varios terminales. Solamente uno de los sistemas de gestión de base de datos en este libro tiene una verdadera capacidad multiusuario tal como la que se encuentra en los grandes sistemas y minicomputadoras. (El capítulo 2 describe qué es lo que tiene que hacer un DBMS multiusuario).

Los sistemas de gestión de base de datos están limitados principalmente por causa de que las redes no son standard; una red de TeleVideo no tiene el mismo software que una red Ohio Scientific, y ni siquiera se parece a un sistema MP/M, multiusuario Altos. Los proveedores de los sistemas de gestión de base de datos no desean apoyar versiones costumbristas locales y en cambio apuntar a los mercados potenciales. Algunas veces una firma de hardware ofrece un DBMS adaptado a sus propias computadoras, pero su acción excluye al paquete del mercado general del software y deja a los propietarios de computadoras prisioneros de sus compañías de computación.

Con la llegada de los discos rígidos para las microcomputadoras, el tiempo de respuesta es frecuentemente menor que el de las grandes computadoras, y la capacidad de almacenamiento es de hasta 8 megabytes por archivo con un CP/M de 8 bits. Si su aplicación no requiere compartir la información ó 100 megabytes de almacenamiento en disco, usted puede ahorrar una gran cantidad de dinero utilizando una microcomputadora. Si usted necesita una red modesta, compre una microcomputadora de 16 bit y espere la próxima generación de software de base de datos.

TERMINOLOGIA

Los términos utilizados en la industria del software originan los manuales del usuario de software. Como era de esperarse, hay muy poca concordancia y acuerdo con la terminología de la base de datos. Esta sección establece como se utilizarán los términos en esta sección o libro que le suministraremos. Los términos se utilizarán consistentemente, aún cuando los vendedores describan sus sistemas con otros términos. Si estas definiciones le parecen confusas al principio, no se preocupe; se volverán cada vez más claras en la medida en que lea

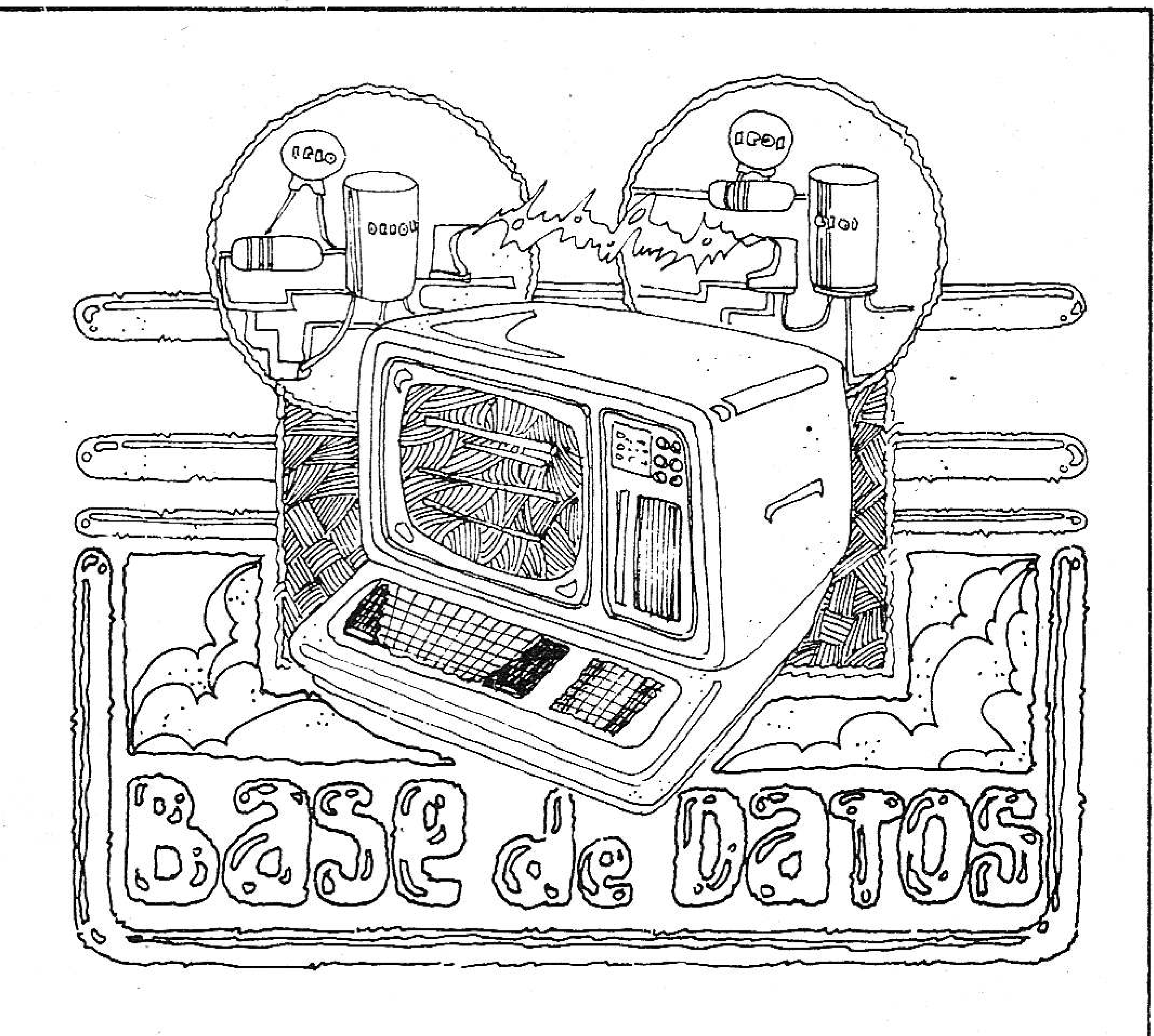
SISTEMA DE GESTION DE BASE DE DATOS (DBMS)

Un DBMS es un paquete de programas y documentación de computación que le permite preparar o "poner a punto" y utilizar una base de datos.

BASE DE DATOS

Una base de datos es una colección de información interrelacionada. Específicamente, la base de datos es una colección de información (datos), indicadores, tablas, índices, diccionarios, etc. Si usted mantiene un registro de sus clientes y productos, estos son parte de la misma base de datos, a menos que usted nunca tuvo la intención de relacionar ambos en ninguna forma.





BASEDATOS

No confundir el término de una sóla palabra con el término base de datos de tres palabras. Una Basedatos es una parte de la Base de Datos. Considérelo como una tabla bidimensional como la que se ilustra en la Tabla 1-1. Las filas son registros y cada columna es un campo. La tabla usualmente se almacena en disco como un archivo. Un diccionario describe todos los campos y uno o más índices permite(n) un rápido acceso a cualquier registro. La tabla, el diccionario y los índices son todos considerados como parte de la basedatos. Una lista de compradores y una lista de productos serían dos basedatos. No todas las bases de datos de los vendedores se pueden separar en basedatos.

ARCHIVO

Como en CP/M, un archivo es una colección de información (datos) en un disco al que se accede por un único nombre. Generalmente contiene una secuencia de registros de formato idéntico, cada uno de ellos contiene una serie de campos. En cambio puede contener un diccionario de datos, un índice, una disposición en pantalla, o cualquier combinación de lo antes mencionado.

REGISTRO

Un registro es un grupo de campos relacionados de información tratados como una unidad. Cada fila en la Tabla 1-1 es un registro.

CAMPO

Un campo identifica una ubicación en un registro donde se almacena un item de información. Este campo tiene ciertas características, tales como longitud y tipo (hilera numérica o de caracteres) y se lo representa por una columna en la Tabla 1-1.

ITEM DE INFORMACION (DATOS)

Un item de información (datos) es otro nombre para un campo utilizado conjuntamente con grandes sistemas DBMS y microcomputadores con red DBMS.

BYTE

Un byte es la parte más pequeña de información en la base de datos, de interés para el usuario. Normalmente equivale a un caracter, pero algunas veces consiste en dos dígitos numéricos. Un megabyte es un millón de bytes y se lo utiliza más rápido de lo que usted puede pensar.

ESQUEMA/PLAN

Un esquema/plan es una figura lógica de la base de datos que ilustra la relación entre varios registros. Los contenidos detallados de los registros provienen del diccionario de datos/información. Los esquemas/planes se utilizan más adelante para ilustrar los conceptos esenciales de la base de datos.

DICCIONARIO DE DATOS/INFORMACION

El diccionario de datos/información es una descripción completa de los campos en una basedatos o base de datos. El diccionario de datos/información describe las relaciones entre los diferentes campos en una base de datos. También describe a cada campo por su nombre, símbolo identificador principal de informe, longitud, tipo de información/dato, límites altos y bajos, etc.

INDICE

Un índice es una tabla de números de registro. Estos números de registro, denominados **Indicadores** están dispuestos para ayudarle a encontrar registros particulares en forma rápida mediante pulsación. El índice permite recuperar los registros en orden secuencial y permite la inserción de nuevos registros. Pero tenga cuidado. Algunos índices no tienen esta capacidad de "adicionar".

TECLA

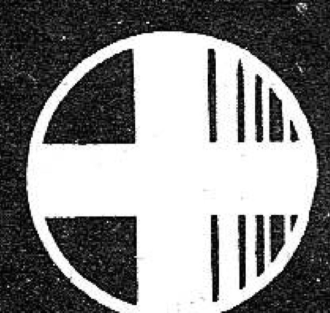
Una tecla es un identificador único para un registro. Puede ser un campo único o un grupo de campos.

TABLA 1-1 Registros y Campos

	Campo	Campo	Campo	Campo	Campo
Registro 1	Nombre	Calle	Ciudad	Estado	C. Postal
Registro 2	Nombre	Calle	Ciudad	Estado	C. Postal
Registro 3	Nombre	Calle	Ciudad	Estado	C. Postal
Registro 4	Nombre	Calle	Ciudad	Estado	C. Postal

LA TECNOLOGIA AL SERVICIO DE LA SALUD





Centro Médico Santa Cruz

EL MAS MODERNO CENTRO DE COMPUTOS AL SERVICIO DEL PACIENTE

Av. SAN JUAN 2483 - Tel. 942-1751 y 941-0055

Soluciones mínimas a los dilemas lógicos

Mientras que existe más para simplificar ecuaciones lógicas que ABC, los diseñadores deben recordar que las funciones lógicas minimizadas no siempre implican configuraciones mínimas del hardware.

McCoy es el escritor del artículo "El ABC de la simplificación de las ecuaciones lógicas, en forma simple".

Para trabajar adecuadamente con las operaciones y resultados en un sistema lógico de valor doble se requiere por lo menos de un sistema de valor triple. Es decir, una variable puede existir en una forma afirmada o real, en una forma negada o falsa o puede haber sido eliminada de una expresión lógica.

Por lo tanto, en la siguiente función de 3 variables
f(A, B, C,) = ABC + ABC + ABC
una elección lógica de representación numérica para los términos mínimos, podría ser numeros decimales de 3 dígitos con un dígito 0 representando una variable negada y el dígito 1 representando un término afirmado.

McCoy comenzó con ese enfoque llamando a su representación numérica números lógicos binarios (BLNs). No es un punto importante indicar si este es o no un nombre preciso para estas cantidades. Lo que es importante es que estas cantidades numéricas son las representaciones codificadas de los términos mínimos en una función lógica. Cada posición del dígito corresponde a una variable diferente, y el uso de un 1 o un 0 como un valor de dígito representa el estado lógico de la variable. Por lo tanto, los términos en esta función estarían representados por la lista numérica
010,001,101,

Si Ud. trabaja o no con estas representaciones como cantidades decimales o aún binarias depende de la implementación del algoritmo básico seleccionado.

Antes de mirar más detalladamente el algoritmo, considere la función lógica
f(A, B, C,) = AC + BC + ABC
obviamente, se necesita algún método para representar las variables B y A faltantes en los primeros dos términos de la función. Si se eligiera un sistema binario para representar la función, saldrían a la luz los problemas.

No hay una forma conveniente para mantener la representación de las tres variables o sus estados lógicos. Además puede que algunas variables falten o puedan faltar de los términos de la función, siendo necesario recurrir a una segunda lista de cantidades numéricas. Estas cantidades representarían, por ejemplo, aquellas variables que han desaparecido de los términos.

Con el sistema decimal se dispone de otros dígitos más allá del 0 y del 1. Por ejemplo, Ud. puede representar la función recién descripta utilizando el dígito 2 para representar una variable afirmada en una función que tiene dos de tres términos presentes. Un dígito 1 se puede utilizar para indicar que una variable ha desaparecido (si un dígito 2 se encuentra presente en cualquier lugar del término), mientras que el dígito 0 continúa representando una variable negada. Por lo tanto, la lista numérica que representa la función de muestra se transforma en
210, 102, 100

La elección de cualquiera de estas representaciones depende obviamente del método numérico desarrollado para implementar el algoritmo.

Representación de una función lógica en el espacio

Resulta posible representar una función lógica de tres variables en la forma de un cubo tridimensional (tambien denominado un cubo-3).

Un cubo-3 tiene 8 vértices, y coincidentemente una función lógica de tres variables puede tener hasta 8 términos mínimos únicos. En general, una función lógica de n variables puede representarse en un espacio n-dimensional por la totalidad o parte de un cubo-n. Un cubo-n tiene 2^n vértices que pueden representar todos los términos mínimos 2^n posibles en cualquier función lógica de n variables.

La figura 1 es una representación del espacio lógico para una función de tres variables e indica las representaciones lógicas y numéricas de la totalidad de los 8 términos mínimos posibles.

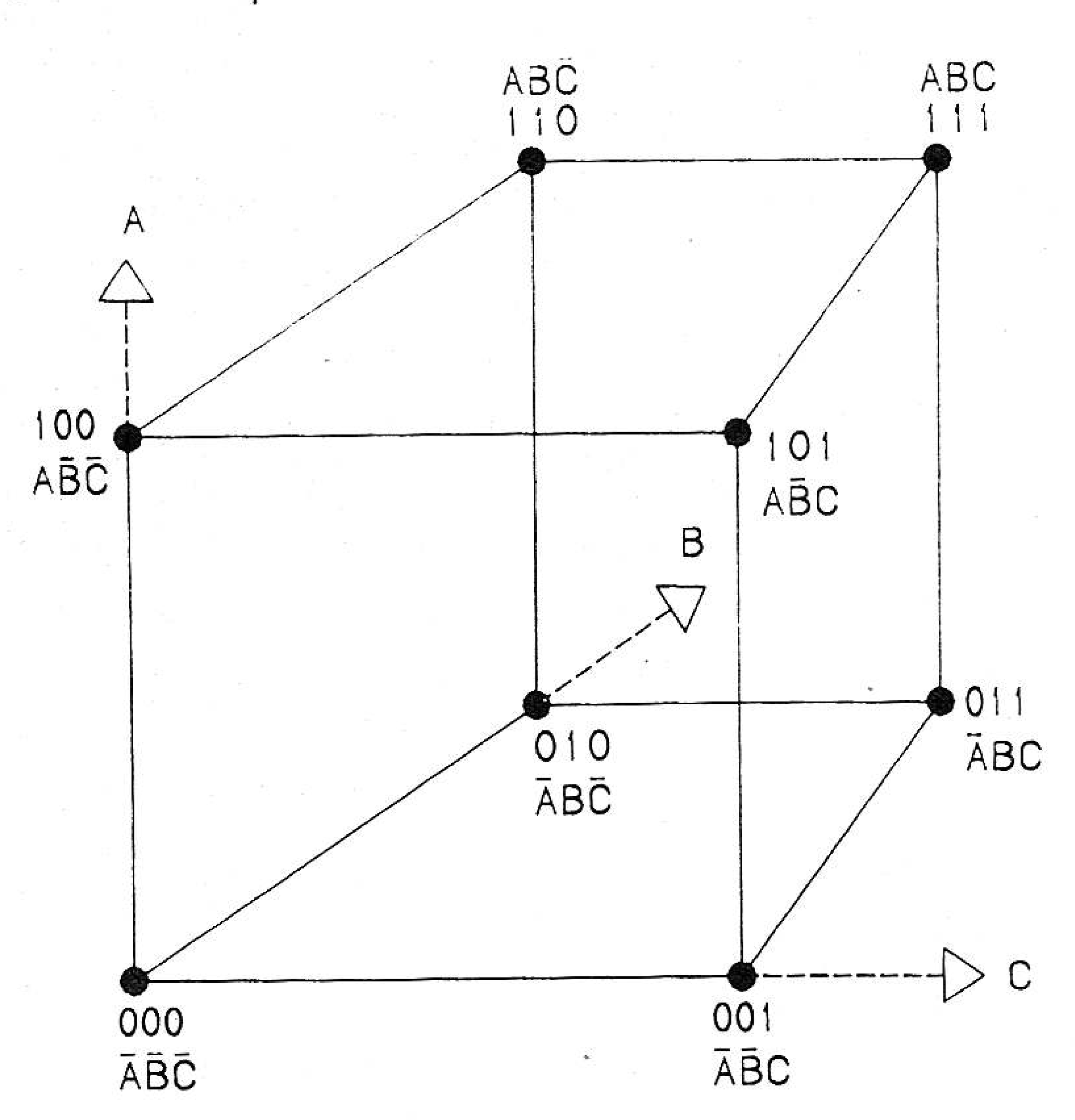


Fig. 1: representación triespacial de los 8 términos mínimos de una función lógica de 3 variables.

El algoritmo topológico para simplificar las funciones lógicas, llamado el algoritmo Star (estrella) indica que dos términos mínimos (vértices del cubo-n) puede ser cubierto por un cubo-1 si están o son adyacentes o colineales.

En otras palabras, los pares vértices colineales pueden tener una línea trazada entre ellos que representa el nivel siguiente más alto o mayor de simplificación de la función. Un examen más profundo de la figura 1 indica que cualquier par vértice adyacente o colineal difiere de otro par sólo por una variable. Tal como se indica mediante las representaciones numéricas de estos términos mínimos, los pares vértices colineales tienen una diferencia aritmética decimal de 1, 10 ó 100 en una función de tres variables. Es una función de 4 variables (espacio-4) los pares vértices colineales tienen diferencias aritméticas decimales de 1, 10, 100, 1000, y así sucesivamente.

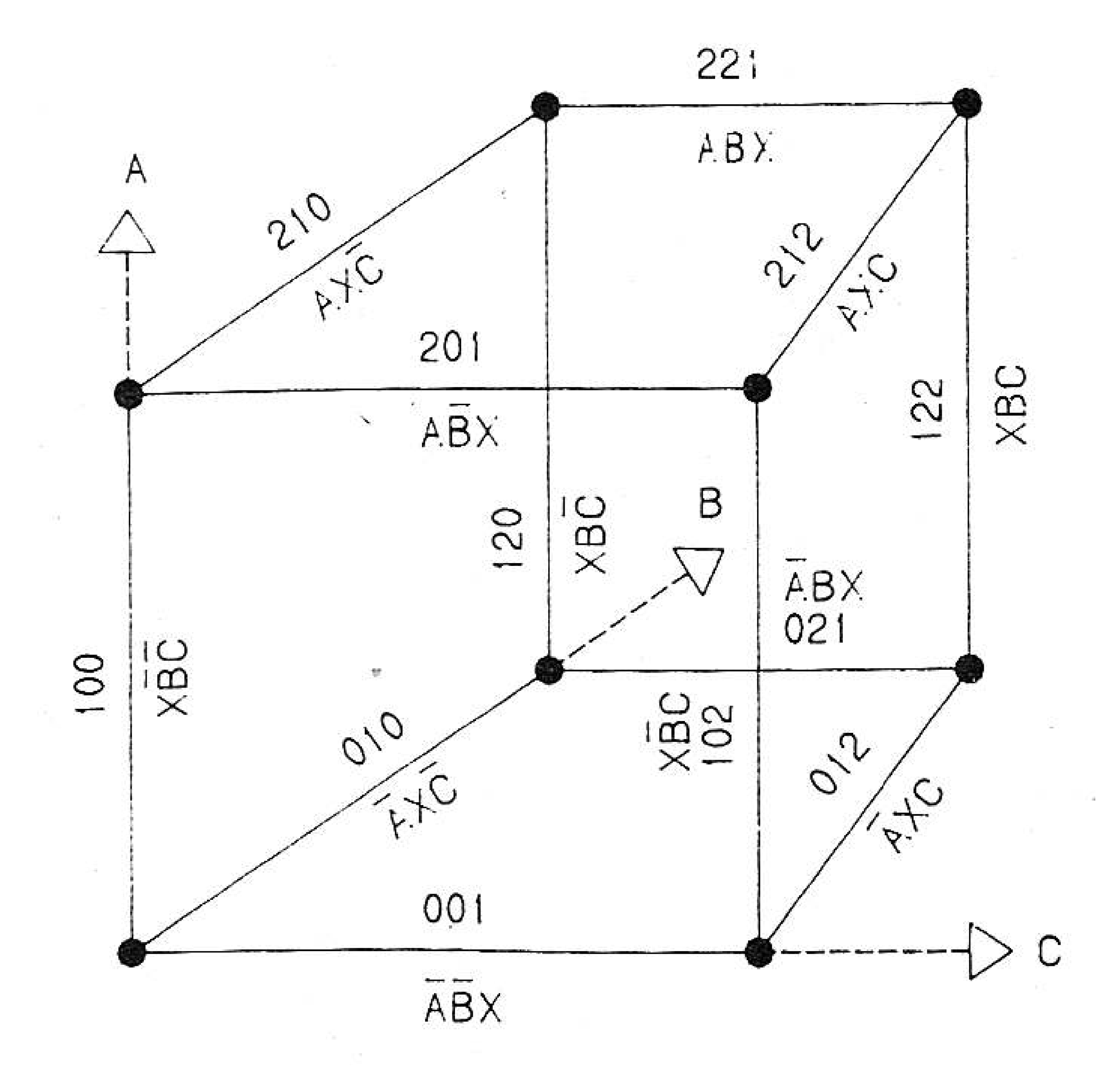
El 1 en el término de diferencia representa la variable lógica que desaparece en el siguiente nivel de simplificación - la línea construida entre los pares vértices de acuerdo con el algoritmo de simplificación.

Ahora, una representación aritmética para el cubo del próximo orden superior (los cubos-1) deberá ser determinada. En el par vértice colineal (110, 111) el término de diferencia es 001 y el término de suma es 221. Habiendo ahora pasado al siguiente nivel de simplificación, y también a la siguiente dimensión superior en el espacio-3, el dígito 2 es apropiado para representar las variables afirmadas en este nivel. Es también apropiado que un dígito 1 represente la variable faltante. Por lo tanto, el término 221 se transforma en ABX, donde X indica simplemente que la variable C ha desaparecido en la simplificación. en forma similar, las representaciones numéricas de los cubos-1 en la tabla 1 pueden determinarse directamente examinando las sumas de sus vértices componentes.

Tabla 1

Representación numérica de
f(A,B,C,) = ABC + ABC + ABC + ABC + ABC + ABC

Par vértice	Diferencia	Suma	Representación lógica
(100,110)	010	210	AC
(001,011)	010	012	AC
(000,001)	001	001	AB
(000,010)	010	010	AC



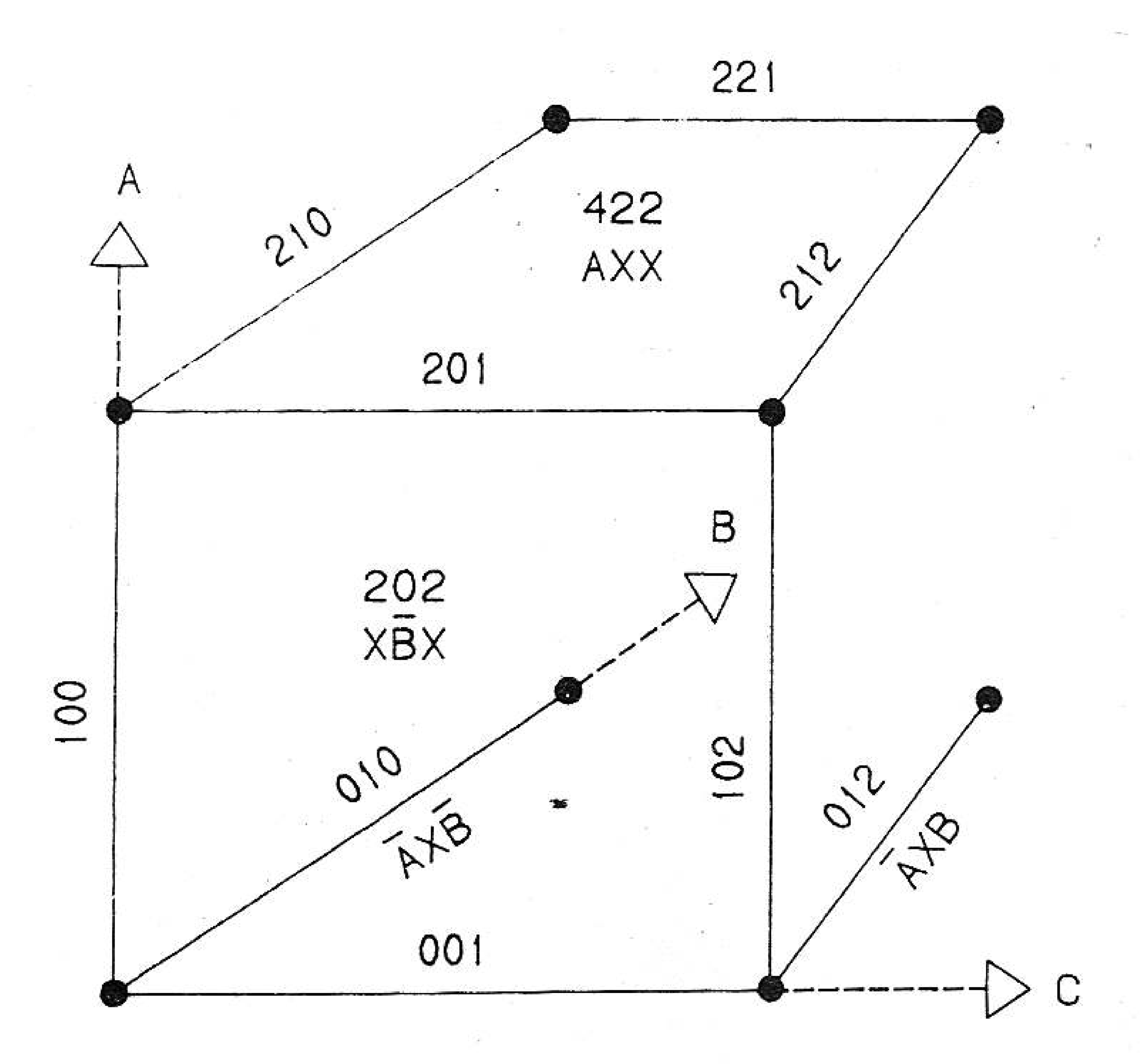


Fig. 2: representación triespacial de los 12 cubos-1 que se pueden formar de los 8 términos mínimos posibles de una función lógica de tres variables.

Fig. 3: representación triespacial indicando la combinación de cubros-1 coplanares para formar superficies.

La figura 2 ilustra las representaciones numéricas y lógicas de la totalidad de los 12 posibles cubos-1 que se pueden construir desde los 8 vértices en un espacio-3. Note que ahora es necesario seguir el rastro del nivel dimensional de simplificación al que Ud. se ha movido. En otras palabras, los términos de suma de las últimas dos entradas en la tabla 1, 001 y 010 se distinguen de los términos mínimo y de los vértices-0. Ud. sólo sabe que el término se generó de la combinación de dos vértices y por lo tanto debe ser una representación de un cubo-1. Parte de la implementación del algoritmo por lo tanto debe incluir el hecho de mantener listas de cubos separados generados en cada dimensión.

En el siguiente nivel de simplificación de algoritmo, Ud. debe buscar juegos de cubos-1 y buscar pares coplanares —pares de cubos-1 que yacen en lados opuestos del mismo plano que podrían ser lados opuestos de la misma superficie—.

Dos pares coplanares de cubos-1 —4 en total—, son necesarios para permitir la construcción de una superficie o cubo-1 que cubra la totalidad de los 4 cubos-1.

La figura 3 ilustra varios cubos-1 seleccionados y dos superficies construidas o cubos-2. Los pares cubo-1 (210, 212) y (201, 221) son coplanares y forman una superficie. En forma similar, los pares cubo-1 (001, 201) y (100, 102) son coplanares y forman otra superficie. El par cubo-1 (010, 012) también es coplanar, pero forma solamente la mitad de una superficie. La técnica de suma y diferencia anteriormente descripta, también se puede utilizar aquí para descubrir pares coplanares de cubos-1, tal como se ilustra en la tabla 2.

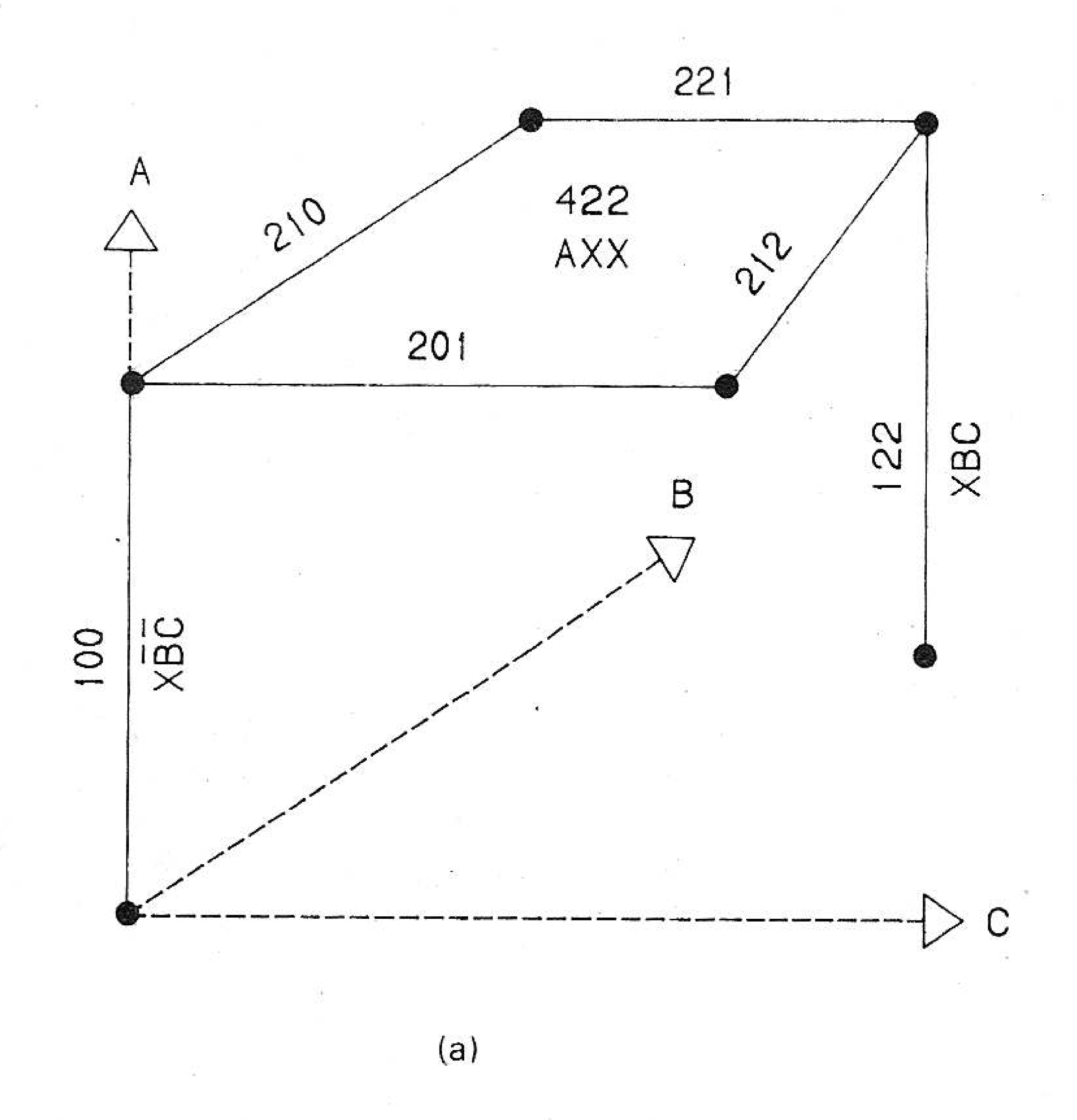
Tabla 2

Generación numérica de cubos-2 a partir de cubos-1 descubiertos en la tabla 1.

Par vértice	Diferencia	Suma	Representación lógica
(210,212)	002	422	A
(201,221)	020	422	A
(001,201)	200	202	B
(100,102)	002	202	B
(010,012)	002	022	A

Los términos de diferencia que indican los cubos-1 coplanares son 002, 020 y 200, respectivamente. En forma similar, las diferencias en un espacio-4 entre los cubos-1 coplanares sería 0002, 0020, 0200 y 2000. Los términos de suma de los pares de cubos-1 coplanares son los mismos si es posible formar una superficie o cubo-2 a partir de ellos.

Por lo tanto, se pueden formar dos superficies a partir de los cubos-1 en la tabla 2: 422 y 202 respectivamente. La aparición de un dígito 4 indica un movimiento al siguiente nivel o dimensión y ahora representa la variable afirmada. Por supuesto, el dígito 0 aún representa una variable negada, y los 2 representan las variables que han sido eliminadas del término. El par cubo-1 (010, 012) puede utilizarse para generar el 022 superficial, si el par coplanar correspondiente (001, 021) está presente. Dado que no está, 012 y 010 de los cubos-1 representan la mejor simplificación posible para aquellos términos particulares.



(a)

Fig. 4: representación triespacial (a) y minimización [(b), (c)] de la función f(A,B,C) = ABC + ABC + + ABC + ABC + ABC + ABC + ABC

Implementación del algoritmo

Al expresar un método numérico para implementar el algoritmo, el primer paso es encontrar un esquema que permita el fácil descubrimiento de cubos-n que pueden estar cubiertos por la construcción de cubos (n + 1) a partir de ellos. La figura 4 (a) representa una función lógica de tres variables alrededor de la cual se debe desarrollar la implementación. Si los vértices se estructuran en orden numérico ascendente, todos los pares adyacentes se pueden descubrir simplemente restando cada vértice de orden superior del de orden inferior y observar las diferencias 001, 010, 100 que indican vértices adyacentes tal como se ilustran en la figura 4 (b). Por lo tanto, la lista ordenada de cubos-0 es

cubo-0	Término Mínimo
000	ABC
011	ABC
100	ABC
101	ABC
110	ABC
111	ABC

No se encontró ningún cubo-0 descubierto cuando se rastreó la lista de cubo-0 cubiertos en la figura 4 (b). Por lo tanto, Ud. debe pasar al siguiente nivel para buscar pares cubo-1 coplanares.

Ordenando la lista de cubos-1 descubiertos en secuencia ascendente, da como resultado lo siguiente:

Cubo-1	**Término Lógico**
100	BC
122	BC
201	AB
210	AC
212	AC
221	AB

Se utiliza la misma técnica para realizar la búsqueda de pares coplanares de cubos-1 tal como se ilustra en la figura 4 (c). Revisando los resultados en la figura 4 (c) se revela un cubo-2, 422, que ha cubierto los cubos-1, 201, 221, 210 y 212. Esto deja a los cubos-1 100 y 122 al descubierto. La forma final de función está representada por las listas

Cubo-0	**Cubos-1**	**Cubos-2**
Ninguno	100 (BC)	422 (A)
	122 (BC)	

Traduciendo las listas nuevamente a términos lógicos:

$$f(A,B,C,) = BC + BC + A$$

Descubrimiento de cubos-1 en función

Par	**Diferencia**	**Vértice 000 Suma (cubo-1)**	**Cubos-0 cubiertos**
(000,011)	011	No colineal	
(000,100)	100	100	000,100
(000,101)	101	No colineal	
(000,110)	110	No colineal	
(000,111)	111	No colineal	
		Vértice 011	
(011,100)	089	No colineal	
(011,101)	090	No colineal	
(011,110)	099	No colineal	
(011,111)	100	121	011,111
		Vértice 100	
(100,101)	001	201	100,101
(100,110)	010	210	100,110
(100,111)	010	212	101,111
		Vértice 110	
(110,111)	001	221	110;11

(b)

Descubrimiento de cubos-2 en función

Par	**Diferencia**	**Vértice 100 Suma (cubo-2)**	**Cubos-1 cubiertos**
(100,122)	022	No coplanar	
(100,201)	101	No coplanar	
(100,210)	110	No coplanar	
(100,212)	112	No coplanar	
(100,221)	121	No coplanar	
		Vértice 122	
(122,201)	079	No coplanar	
(122,210)	088	No coplanar	
(122,212)	090	No coplanar	
(122,221)	099	No coplanar	
		Vértice 201	
(201,210)	009	No coplanar	
(201,212)	011	No coplanar	
(201,221)	020	422	201,221
		Vértice 210	
(210,212)	002	422	210,212
(210,221)	011	No coplanar	
Vértice 212			
(212,221)	009	No coplanar	

(c)

Garantizando una función mínima

En respuesta al artículo de McCoy, Guerrero seleccionó un ejemplo que reclama que no funciona con el algoritmo de McCoy. En respuesta a los reclamos de Guerrero, McCoy indicó que él no dijo que el algoritmo produce una función mínima absoluta. Tal como se evidenció en los ejemplos anteriores desde los cuales se desarrolló la implementación, el método encuentra **todos** los cubos de orden superior que pueden construirse de cubos de orden inferior, aún cuando los cubos de orden inferior ya estaban cubiertos. Aún la naturaleza topológica del algoritmo requiere que los cubos de orden superior sean construidos para descubrir si es posible una minimización posterior.

Se requiere un paso final para seleccionar la solución mínima luego de haberse construido todos los cubos posibles de orden superior.

La solución se debe seleccionar de la lista de cubos, los de orden superior primero, y dicha selección debe detenerse si la totalidad de los cubos (n-1) han sido cubiertos por la selección de cubos-n. La inclusión de cubos (n-1) en tal caso simplemente adiciona términos redundantes que ya están expresados por los cubos-n. Esto no es necesariamente una estrategia de minimización mala o indebida. El hecho es que la minimización de una función lógica en un medio ambiente de diseño es una estrategia que no siempre resulta en el mínimo absoluto.

La función que Guerrero indicó como un caso imposible de realizar se expresa en el espacio-3 ilustrado en la figura 5(a). Nótese que los tres cubos-1 pueden construirse desde los tres vértices. Estos es en realidad una versión minimizada de la función, pero solamente se requieren los cubos-1 120 y 212 para cubrir la totalidad de los tres vértices y producir una función mínima absoluta. Para construir una minimización absoluta en el algoritmo se necesita una búsqueda de cubos-0 cubiertos (ver figura 5(b)).

No existen más cubos de orden superior que se puedan construir a partir de la recolección de los cubos-1 que sobran; por lo tanto se ha alcanzado el fin de la minimización posible. Para lograr seleccionar una representación mínima de la función, la lista final de cubos-1 debe ser rastreada conjuntamente con la lista de cubos-0 originales hasta que no quede ninguno tal como se ilustra en la figura 5(c).

Los primeros dos cubos-1 cubren la totalidad de los términos mínimos originales y son una representación mínima de esa función. Si este paso ha sido realizado por el programa calculador ofrecido por McCoy, el resultado habría sido una solución mínima real. La función minimizada es entonces
f(A, B, C) = BC + AC

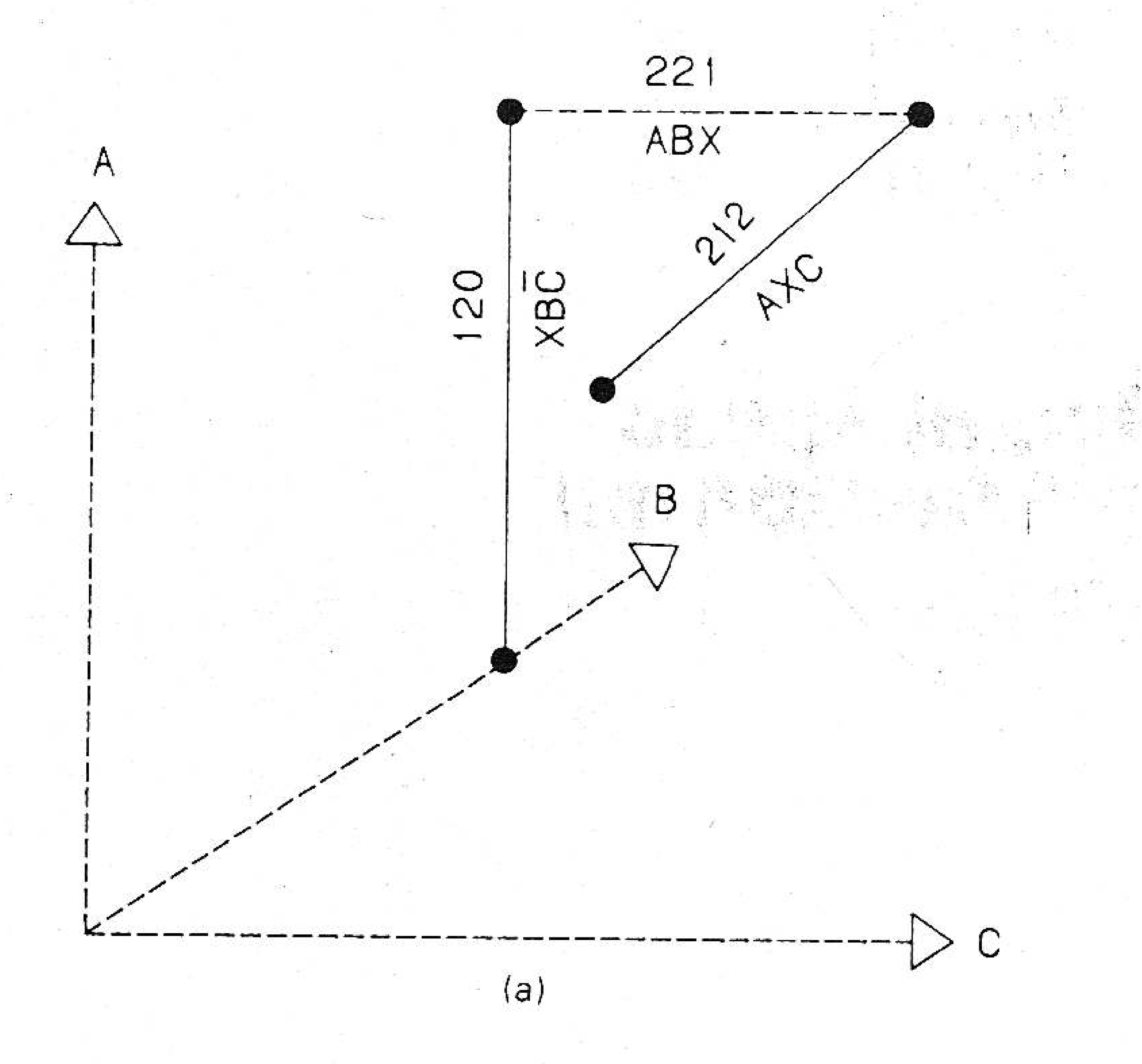


Fig. 5: representación triespacial (a) y minimización (b), (c) de la función F(A,B,C) = ABC + ABC + ABC + ABC

Pasos de minimización

Par	Diferencia	Suma (cubo-1)	Cubos-0 cubiertos
	Primer paso		
(010,101)	091	No colineal	
(010,101)	100	120	010,110
(010,111)	101	No colineal	
	Segundo paso		
(101,110)	009	No colineal	
(101,111)	010	212	101,111
	Tercer paso		
(110,111)	001	221	110,111

(b)

Paso final

Cubos-1	Cubos-0 cubiertos	Cubos-0 restantes
		010,101,110,111 (cubos-0 originales)
120	010,110	101,111
212	101,111	ninguno
221	110,111	solución redundante

(c)

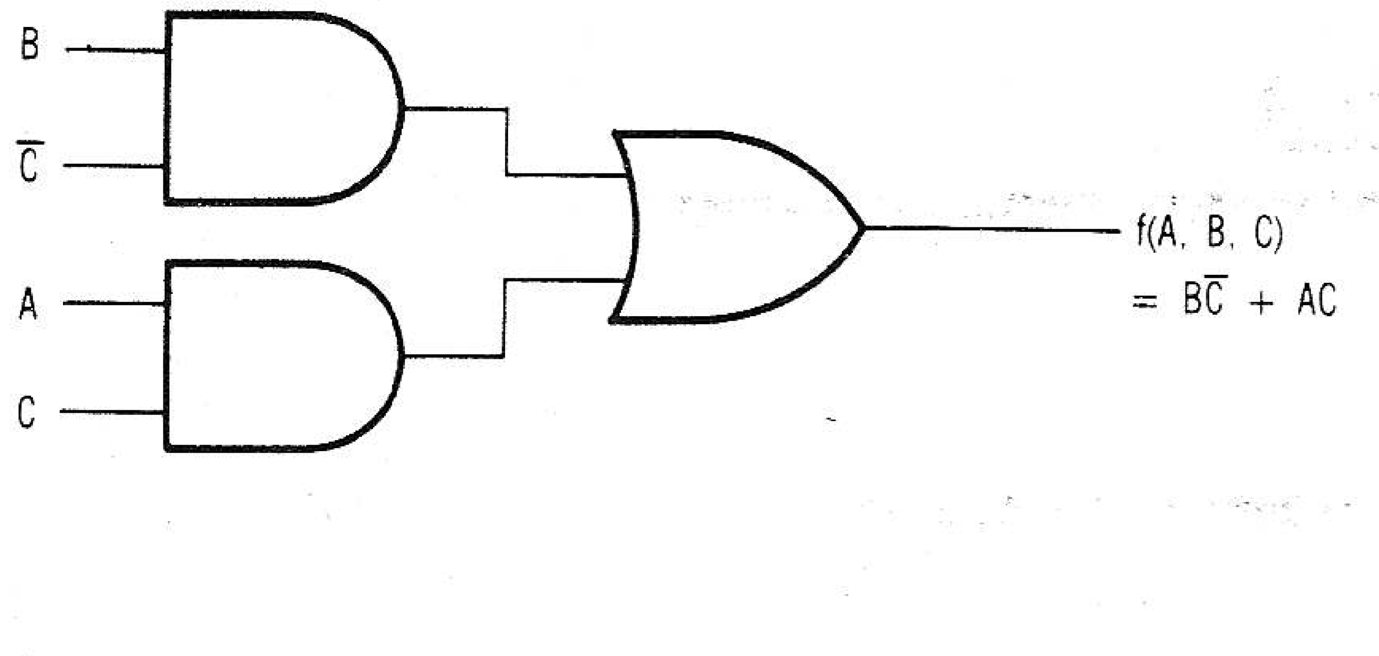
Los programas siempre tienen errores

Alguien alguna vez dijo que ningún programa se terminaba sin tener errores; esto solamente se acerca al estado libre de errores. Aunque McCoy intentó mostrar que se puede desarrollar una técnica de minimización para un medio ambiente programable, su método tiene ciertos errores. Su programa realmente produce los resultados correctos para sus propios ejemplos y aún para el término redundante mencionado por Guerrero. Sin embargo, no funciona para los casos donde la función mínima tiene cubos de órdenes diferentes en el resultado final. Por ejemplo, alimentar la información desde el ejemplo en la figura 4 al programa producirá un sólo cubo-2 A como la respuesta. El método pierde completamente los dos cubos-1 no cubiertos por el cubo-2 A, por cuanto McCoy no tomó en consideración los cubos cubiertos en la medida en que se movía al siguiente nivel de minimización (la siguiente dimensión de orden superior). Su programa reemplaza la totalidad de los cubos (n-1) generados a partir de ellos. Por lo tanto, si todos los cubos (n-1) no se cubren por los cubos-n, se pierde parte del resultado final. En resumen, el concepto propuesto por McCoy funciona, aunque el programa carece de la lógica necesaria para completar la implementación.

Para corregir el programa, se debe mantener una lista de cubos adicional de manera que se pueda verificar la cobertura del cubo (n-1) y se pueda retener los cubos no cubiertos luego de cada nivel de mimización. Los cubos-0 cubiertos conjuntamente con la lista de cubos-1. Luego de formarse una lista completa de cubos-1, la lista de cubos-0 cubiertos debe compararse con la lista de cubos-0 originales y la totalidad de los cubos-0 que no están en la lista cubierta deben retenerse en una lista separada. La lista de cubos-1 recientemente generada se transforma en la nueva "lista de entrada" a ser buscada para descubrir los cubos-2 que se pueden formar, y se puede descartar la lista de cubos-0 cubiertos. Nuevamente se debe mantener una lista de cubos-1 cubiertos generados de la lista de cubos-2 y luego de completarse el proceso, se forma y retiene una lista remanente que consiste de cubos-1 no cubiertos. Suministrar una versión de trabajo del programa, es un hecho que está mucho más allá del alcance de este artículo.

Lo mínimo no es siempre lo mínimo

Para asumir que la minimización de una función lógica siempre produce una implementación mínima de hardware no es siempre válido. En los días de las implementaciones lógicas del componente discreto, las técnicas de minimización se mantenían con una alta reputación dado los esfuerzos inacabables para reducir el costo general del hardware.



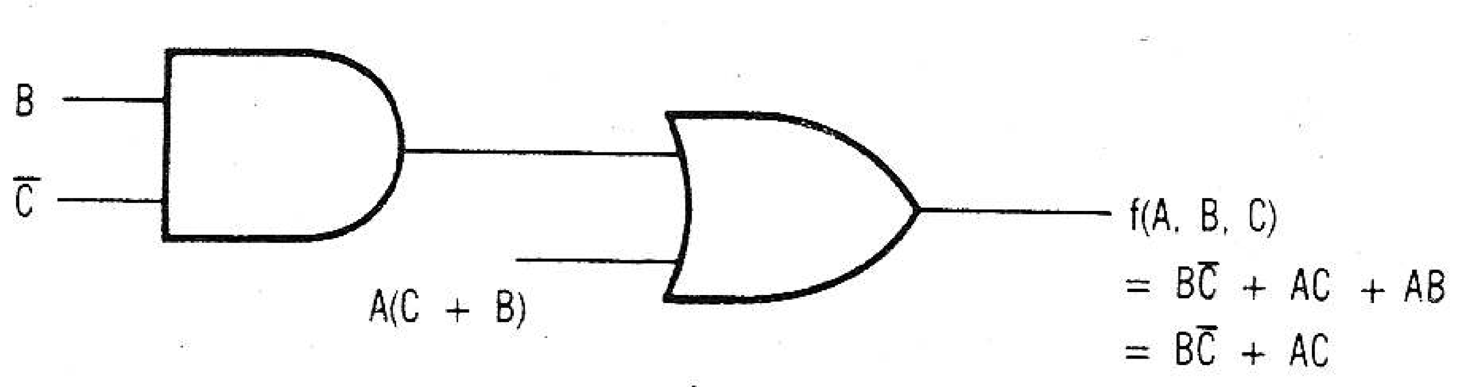


Fig. 6: implementación de la función en la figura 5.

usted mismo", los cuales permitieron al diseñador de lógica participar nuevamente en el nivel de puerta del diseño lógico, es bien cierto que el interés por las técnicas de minimización volvió a resucitar.

Cuando se practica el arte de la minimización la solución mínima absoluta en los términos de función lógica no es siempre deseable.

En realidad, la aseveración de McCoy que un lógico competente debería poder reconocer una solución mínima, es parcialmente correcto. Algunas veces, una solución intermedia, una que contenga deliberadamente términos redundantes, es realmente la más simple desde el punto de vista del hardware.

Considérese, por ejemplo, las dos versiones de la solución para la función en la figura 6. Si el mínimo absoluto es el empleado, se requieren tres puertas. Si ya se dispone de una parte de la función no tan mínima como una señal lógica, sin embargo dará un buen resultado una implementación de 2 puertas. Aún peor es el caso donde la función de muestra se implementa en una plaqueta que tiene solamente las señales B, C y la señal compuesta A (C + B) disponible en las entradas de la plaqueta. En este caso, para forzar la implementación de la función mínima requeriría la presencia de las señales A,B o C en las entradas de la plaqueta,, que a su vez requiere una entrada adicional a la plaqueta así como una vía para placa de base auxiliar para la señal extra.

Dado que la mayoría del diseño lógico complejo en el nivel de puerta se realiza por división de función, usted debe considerar todas las minimizaciones intermedias posibles de una función dada para asegurarse de que la implementación resultante total del hardware es en realidad mínima. Si el concepto de utilizar uno o más términos redundantes en una implementación es el que se va a utilizar, esto puede conducir a una configuración más económica, particularmente dado que la plaqueta y las conexiones de las patas de la placa de base pueden convertirse en una parte significativa del costo.

Con el advenimiento del circuito integrado y de la progresión de integración a pequeña, media y gran escala que siguió repentinamente, casi toda función compleja que u = diseñador de lógica podía necesitar para un diseño dado estaba disponible en la forma de un chip.

La necesidad de cualquier técnica de minimización sería disminuyó rápidamente y quedó virtualmente aplastada con la aparición de la matriz o reticulado lógico programable. Sin embargo, con el advenimiento de la tecnología de la matriz o disposición de compuertas (puertas), las fundiciones de sílice, y los sistemas de fabricación con ayuda de la computadora y los diseños con ayuda de computadoras "para que todo lo haga

El crecimiento de la familia de los procesadores aumenta las opciones del sistema

La expansión de la familia 68000 para incluir otro procesador y chips periféricos libera a los diseñadores del sistema de las limitaciones de la aplicación

Desde su introducción en 1979, el microprocesador MC 68000 ha sido ampliamente utilizado en aplicaciones que van desde microcomputadoras poco sofisticadas hasta grandes computadoras centrales multiusuario que operan sin detenerse.

El diseño de la MC 68000 de 16 bits interior/32 bits exterior suministra un alto rendimiento a un precio razonable. Un típico problema es que los últimos diseños deben ofrecer siempre un rendimiento mayor o disminuir los costos. Para hacer que las cosas mejoren, los esfuezos de rediseño deben ser mínimos. La última solución para esta dicotomía de diseño yace con el vendedor de microprocesadores que ofrece productos compatibles con códigos objeto que satisfacen ambos requerimientos.

El árbol de la familia del microprocesador M68000 se planeó con la estrategia en mente (Ver figura 1). Los miembros de la familia recientemente introducidos extienden las características tanto hacia arriba o hacia abajo en términos de rendimiento y de costos. A modo de ejemplo, dado que el procesador MC68008 es compatible con el MC 68000, los programas para los sistemas de nivel de entrada pueden conectarse a la MC68008 sin ningún cambio. Este procesador establece el extremo inferior de la familia del procesador M68000 y es básicamente idéntico con el MC68000, excepto que ha sido rediseñado para trabajar con una línea de datos de 8 bits y espacio reducido de direccionamiento.

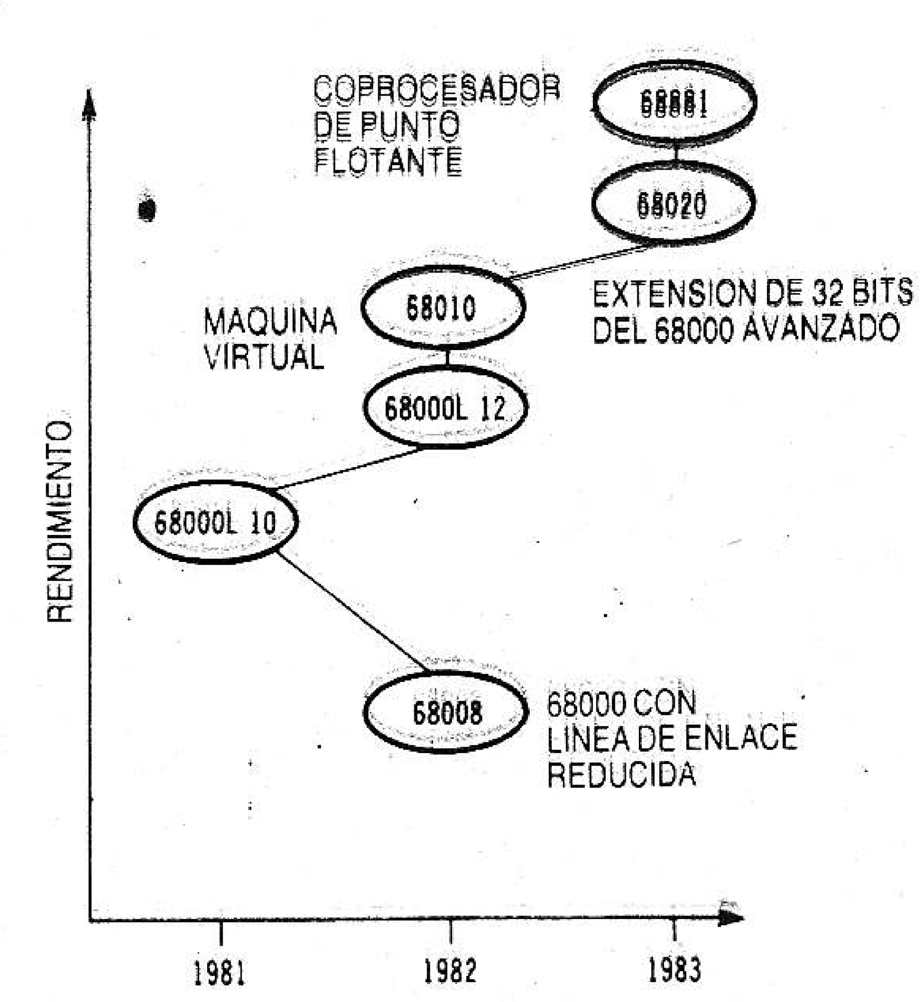


Fig. 1. Genealogía de la familia 68000. Extendiendo los ofrecimientos de la familia tanto hacia arriba como hacia abajo, el fabricante ofrece a los diseñadores rendimiento y bajo costo.

Se consideró la multiplexación (multiplexing) de la dirección y los datos, pero los problemas tradicionales de la performance más baja dirigieron la decisión hacia una implementación de no multiplexación. Manteniendo una estructura de enlace no multiplexado,

MC68008 automáticamente tiene un rendimiento del 30 al 50% por sobre los procesadores de la competencia. El espacio de dirección apoyado por este procesador está limitado a 1 Mbyte, compatible con los sistemas poco sofisticados.

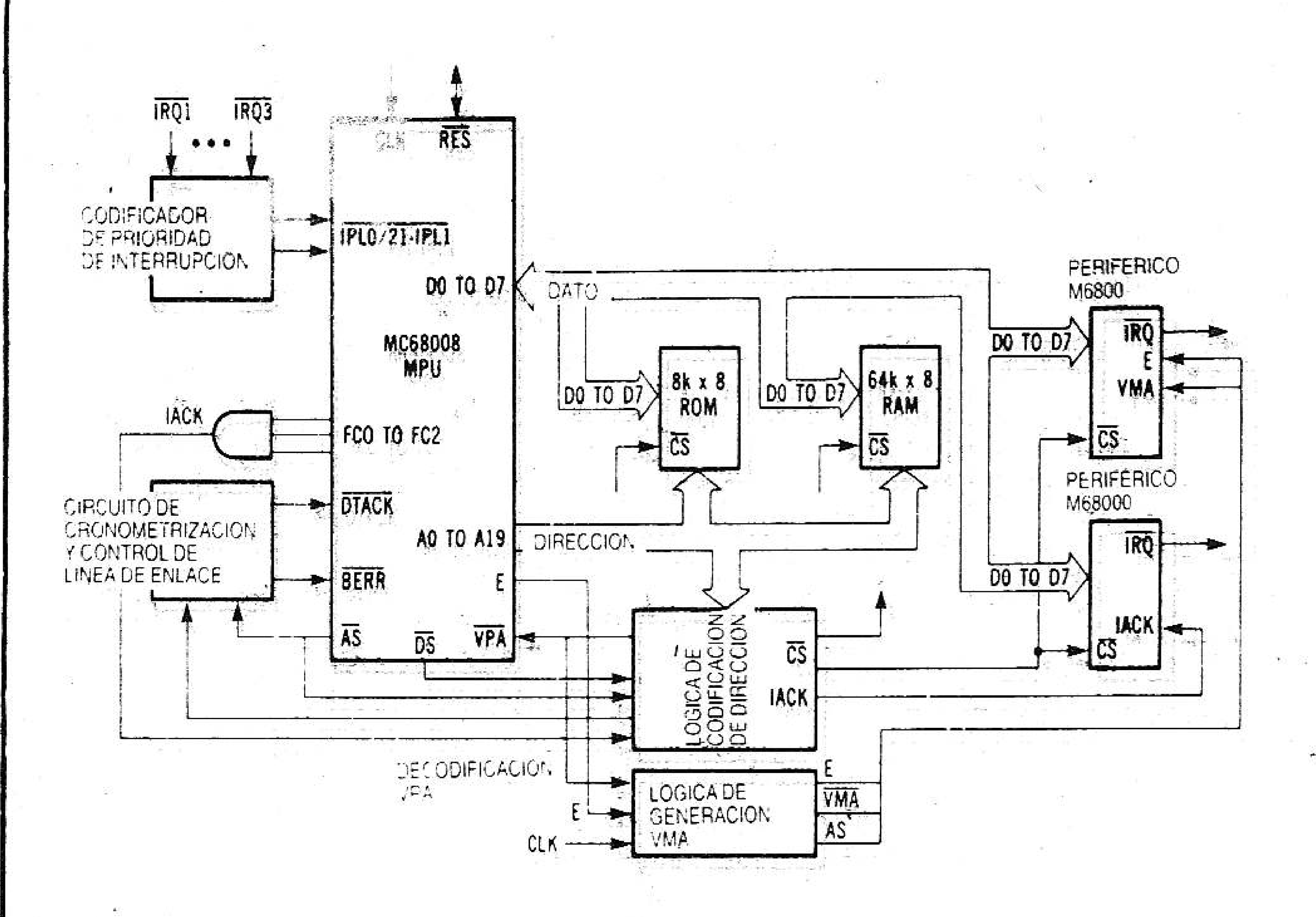


Fig. 2. La configuracion mínima del sistema del procesador MC68008 requiere solamente un ROM de utilidad o provecho. La línea de enlace de datos de 8 bits y el espacio reducido de dirección 61 Mbyte) permite un paquete compacto de 48 conexiones.

La figura 2 ilustra una configuración mínima del sistema del MC68008. Las señales de acuse de recibo operan exactamente igual que para MC68000, una dirección es emitida en el enlace de dirección, se mantiene el estroboscopio de dirección, se emite la información, y se mantiene el estroboscopio de datos. Cuando se completa la operación, como sería en este ejemplo el ciclo de escritura, la señal de reconocimiento de transferencia de datos vuelve al procesador. Esto indica al procesador que el ciclo se ha terminado con éxito y que se pasa a la siguiente operación. La mayoría de las líneas de señal soportadas en el MC68000 también se ofrecen en el MC68008. Las únicas excepciones son los bits superiores de las líneas de datos, la línea de acceso válido a la memoria, una línea de solicitud de interrupción, y la línea de reconocimiento garantizado de enlace. Estas señales se pueden/simular fácilmente en forma externa, pero raramente son necesarias en los sistemas poco sofisticados.

La ventaja principal del MC68008 es que sólo requiere un ROM para el autocomando, RAM de un byte de ancho y puede tomar las ventajas de ambos periféricos M68000 (todos orientados en 8 bits), también algunos periféricos de la familia M68000 (el MC68230 por ejemplo). Dado que la línea de enlace es la de menos de la mitad del ancho y de la dirección de la memoria, el paquete de 48 patas del procesador es más económico.

El paso hacia adelante en la evolución de la familia del procesador 68000 es la máquina virtual MC68010. Este procesador es también código objeto y compatible en lo que a arquitectura se refiere con el MC68000 original. La ventaja principal de la máquina virtual es que recupera el control luego de la falla de la memoria. Por lo tanto, el procesador se puede utilizar en medio ambientes con gran demanda de trabajo y como una máquina virtual con sistemas operativos múltiples funcionando concurrentemente.

Esta máquina con memoria virtual permite a los programas que exceden el espacio de dirección de la máquina básica operar sin la intervención conciente del programador de aplicaciones. Todo el paso de las partes del programa ocurre transparentemente, de manera tal que el usuario nunca sabe que la máquina tiene menos memoria de lo que requiere el programa. Esta capacidad les permite a los sistemas basados en el MC68010 ofrecer características comunes a las máquinas computadoras centrales más grandes, a una fracción de costo.

El MC68010 se diseñó sobre la suposición que la tendencia en los sistemas de computadoras pequeñas se dirigía hacia mayores capacidades de almacenamiento masivo. Este enfoque es posible dado que la mayoría de los programas son lo suficientemente pequeños como para permanecer totalmente en un almacenamiento primario. Por suerte, aquéllos que puedan ocupar totalmente una RAM ocurren en pequeñas proporciones con relación al código general.

Un juego operativo de páginas de memoria puede residir en el almacenamiento primario, mientras que otras partes del programa residen en el almacenamiento secundario (disco).

Cuando se hace referencia a una parte del programa que no está en la memoria primaria, la página requerida se puede ubicar en o quitarse del almacenamiento secundario. Por lo tanto el ciclo de la línea de enlace puede continuar. En lo que al usuario se refiere, el procesador tiene un gran espacio físico de direccionamiento, aunque la cantidad de memoria que se encuentra realmente presente es algo limitada.

La capacidad para soportar la memoria virtual requiere que la referencia de que el procesador detenga una instrucción en el medio de un ciclo de línea de enlace sea enviada nuevamente para que la página sea residente y retorne y continúe la ejecución de la instrucción interrumpida en un plazo posterior. La secuencia de eventos durante el manejo de una falla de la página es el siguiente: luego que se detecta la falla de una página, el MC68010 copia los contenidos de sus registros internos a la pila del supervisor. Esta operación "limpia" la máquina y, luego de una cantidad menor de preparación previa, puede volverse a enviar a otra tarea.

Cuando la tarea anteriormente defectuosa está lista para continuar, el supervisor realiza nuevamente alguna preparación previa, y luego ejecuta una instrucción de retorno de excepción (RTE). La instrucción RTE del MC68010 es más inteligente que la de su predecesor por cuanto permite que el control retorne de las fallas de las páginas así como de las interrupciones. Durante la ejecución de esta instrucción, el procesador recarga todos sus registros temporarios desde la pila del supervisor, y entonces intenta volver a operar el ciclo de la línea de enlace que causó la falla de la página. Cuando la instrucción se completa satisfactoriamente, el programa continúa operando como si nada hubiera ocurrido.

La ventaja obvia del soporte de la máquina virtual es

que, en los sistemas multiusuarios, los programas muy grandes pueden operar en medio ambientes de bajo costo. Otro beneficio es que los programadores de aplicaciones pueden escribir programas muy grandes sin lamentarse por las limitaciones de la memoria. Esto elimina los dolores de cabeza causados por los límites de los segmentos artificiales de memoria o recubrimientos del programa, y simplifica el desarrollo del software.

El siguiente paso más allá de la máquina virtual es el MC68020, una versión de 32 bits del procesador MC68000. Este procesador permite a los programas escritos en el MC68000 operar entre dos y cuatro veces más rápido. Este incremento en el rendimiento se debe al enlace de datos de 32 bits en el MC68020. Dado que el rendimiento del procesador está muy ligado al ancho de banda de la línea de enlace de la memoria, ofreciendo dos veces el ancho de banda, el rendimiento se duplica. Otra característica que aumenta el rendimiento es un esquema de reserva de instrucción. Una vez que un circuito queda atrapado en una reserva o "escondite", el régimen de ejecución es considerablemente más alto de lo que habría sido si se hubieran requerido referencias de memoria externas.

Un compañero del MC68020 es el coprocesador de punto flotante MC68881. Esta extensión de la arquitectura del procesador MC68020 permite una ejecución muy rápida de las instrucciones del punto flotante: aproximadamente 50 veces más rápido que un programa de software cuando está emulando operaciones de multiplicación de punto flotante de 80 bits del IEEE. El coprocesador de punto flotante ejecutará aproximadamente 120.000 operaciones de punto flotante de 80 bits por segundo. El juego de instrucciones del MC68881 también ofrece muchas operaciones aritméticas y trascendentales, todas con un rendimiento igualmente imrpesionante.

Extensiones de la arquitectura del procesador

Los chips de soporte periféricos tienen una influencia directa en el diseño del hardware y software. En la familia del 68000 se encuentran incluidos: la unidad de gestión de memoria MC68451, el controlador de acceso directo a memoria MC68450, y el módulo de arbitraje (arbitral) de línea de enlace MC68452. Cada uno de estos chips periféricos contribuyen a cómo se configurará el sistema último o final. Como resultado, es muy importante que los diseñadores de sistemas comprendan totalmente la filosofía que se encuentra detrás del miembro más recientemente introducido en esta familia, el módulo de arbitraje (arbitral) de línea de enlace MC68452.

Este módulo es un controlador que arbitra en los conflictos entre los masters de las líneas de enlace múltiples cuando todos los controladores solicitan utilizar la línea de enlace al mismo tiempo. Estos controladores pueden ser cualquier combinación de procesadores, controladores de acceso directo a memoria, u otros masters de línea de enlace que necesitan tomar el control de las líneas de dirección y de las líneas de enlace de datos para realizar sus operaciones. El MC 68452 no disminuye el rendimiento del sistema, por cuanto el arbitraje se realiza concurrentemente con la operación del ciclo de la línea de enlace.

El módulo de arbitraje (arbitral) de línea de enlace puede utilizarse en otras configuraciones, por cuanto el chip completa el arbitraje en menos de 50 ns. La familia de masters de línea de enlace MC68000 soporta la simultaneidad de operaciones de la línea de enlace y del arbitraje. Antes de la terminación del ciclo de una línea de enlace, el siguiente master de línea de enlace sabe que el siguiente ciclo de línea de enlace estará disponible para su uso, sin esperar que se lleve a cabo la secuencia de arbitraje.

Las señales responsables del arbitraje son la solicitud de línea de enlace, la concesión de línea de enlace y el reconocimiento de la concesión de la línea de enlace. Durante cualquier ciclo la solicitud de línea de enlace se puede realizar por un periférico e, inmediatamente luego, el procesador enviará nuevamente la concesión de línea de enlace. Esto informa a los periféricos que a la finalización de este ciclo corriente de línea de enlace, puede tener el control de la línea de enlace. El periférico monitorea la señal estroboscópica de dirección del procesador y cuando es negada sabe que el procesador ha liberado la línea de enlace. En este momento, el periférico afirma el reconocimiento de concesión de línea de enlace para indicar que tiene el control de la misma. La liberación de la señal de solicitud de línea de enlace permite al periférico dar el control de la línea de enlace a otros periféricos si existe una solicitud al efecto. Cuando el periférico termina con la línea de enlace, niega la señal de reconocimiento de concesión de línea de enlace y el procesador continúa con sus operaciones de línea de enlace.

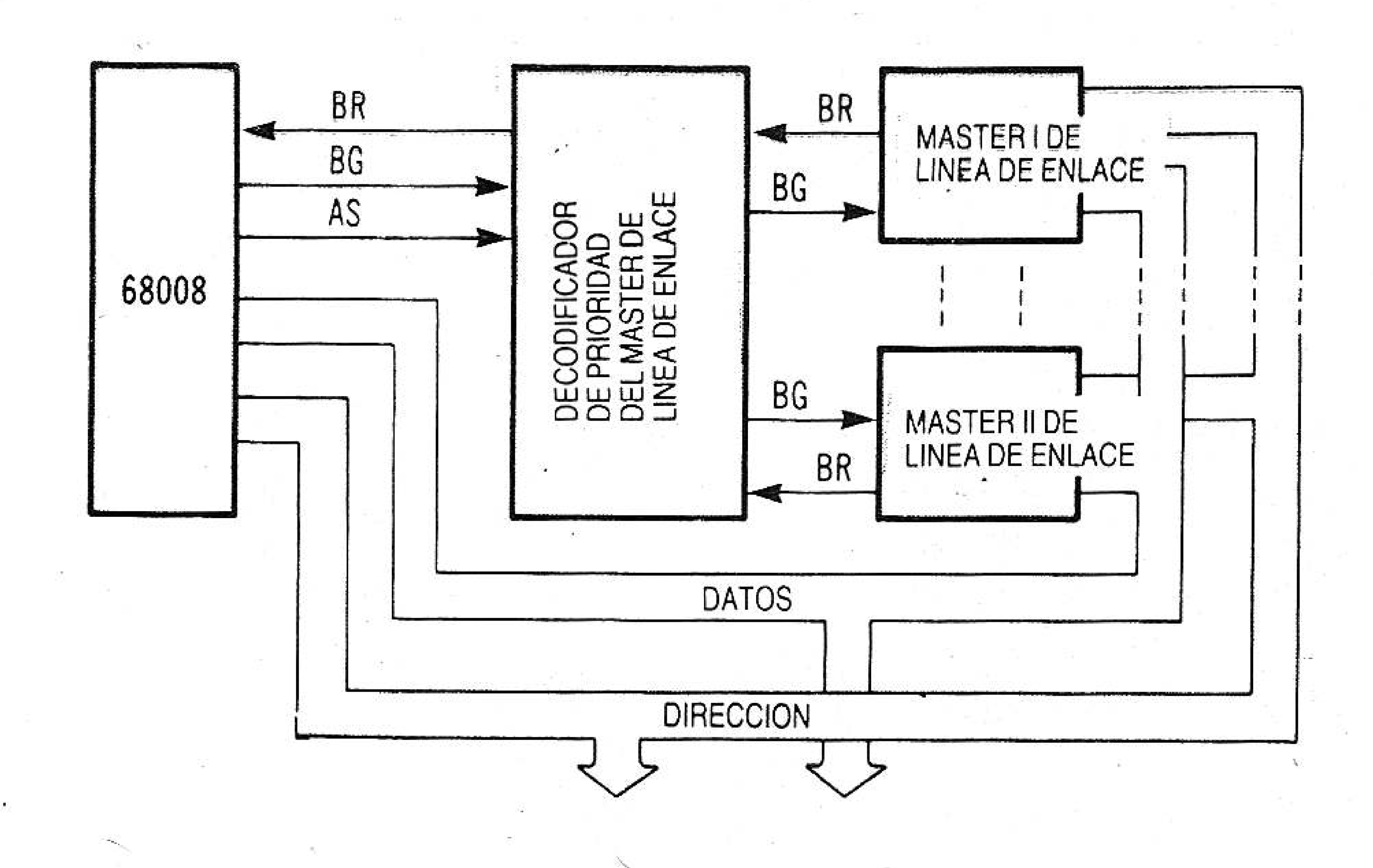


Fig. 3. El módulo de arbitraje de línea de enlace puede dar prioridades a las solicitudes de hasta 7 masters de línea de enlace, basado en un esquema de prioridad física.

La figura 3 ilustra la interconexión entre el módulo de arbitraje (arbitral) de línea de enlace, la unidad de procesamiento central y el controlador de acceso directo a memoria.

Cuando dos masters de línea de enlace de periféricos simultáneamente solicitan el uso de una línea de enlace, el conflicto se resuelve rápidamente por cuanto el rendimiento del sistema decae como resultado del tiempo gastado en el arbitraje. En menos de 50 ns el módulo de arbitraje de línea de enlace puede resolver el conflicto y asignar una prioridad basándose en la ubicación física. Adicionalmente el módulo de arbitraje de línea de enlace emite una señal de liberación de línea de enlace que se puede usar para inhabilitar a los periféricos de prioridad más baja en la utilización de la línea de enlace.

La familia de masters de línea de enlace M68000 soportan en su totalidad el concepto de la re-operación del ciclo de la línea de enlace. Si un dispositivo de prioridad más baja insiste en tomar la línea de enlace, un dispositivo de prioridad más alta puede tomar el control de la línea de enlace antes que el dispositivo egoísta haya completado el ciclo de la línea de enlace que estuvo realizando. Mientras que el dispositivo de prioridad más alta tenga el control de la línea de enlace, el dispositivo de prioridad más baja queda latente. Cuando el dispositivo de prioridad más alta abandona la línea de enlace, el periférico de prioridad baja vuelve a realizar totalmente su ciclo de línea de enlace y continúa como si no hubiera sido interrumpido.

Afortunadamente no todas las partes de los periféricos tienen un impacto tan grande sobre el diseño del hardware y software del procesador central como lo tiene el módulo de arbitraje de línea de enlace MC68452. Existen un número de consideraciones que deben pesarse cuidadosamente, sin embargo, cuando se elija entre tales periféricos de soporte como el controlador periférico inteligente MC68120 y el periférico multifuncional MC68091.

El controlador periférico inteligente MC68120 lleva la capacidad del procesador de entrada/salida a la familia M68000. El controlador periférico inteligente soporta una línea de enlace local de dirección/datos que conecta al sistema con periféricos tales como controladores de discos o interfaces de comunicaciones. La línea de enlace local es sincrónica, opera con un tiempo de ciclo de un microsegundo y trabaja con los periféricos de la familia M68000. Este estilo de entrada/salida permite la interface a periféricos de muy bajo costo que son soportados por muchos vendedores. Adicionalmente, la línea de enlace separada de entrada/salida permite que las operaciones de entrada/salida sean simultáneas con las operaciones de la computadora, por lo que se aumenta la productividad y rendimiento del equipo o sistema.

Un ejemplo del controlador periférico inteligente MC68120 trabajando como un procesador de entrada/salida se ilustra en la Figura 4. Aquí el controlador periférico inteligente se utiliza como un controlador de terminal múltiple.

En este ejemplo, el controlador periférico inteligente se conecta a vías de entrada/salida múltiples en serie en la línea de enlace local a la línea de enlace de instrucción M68000 en el lado del procesador central. La función del controlador periférico inteligente es realizar un almacenamiento intermedio de las vías de entrada/salida en serie y actuar como una vía múltiple primera en salir - primera en entrar.

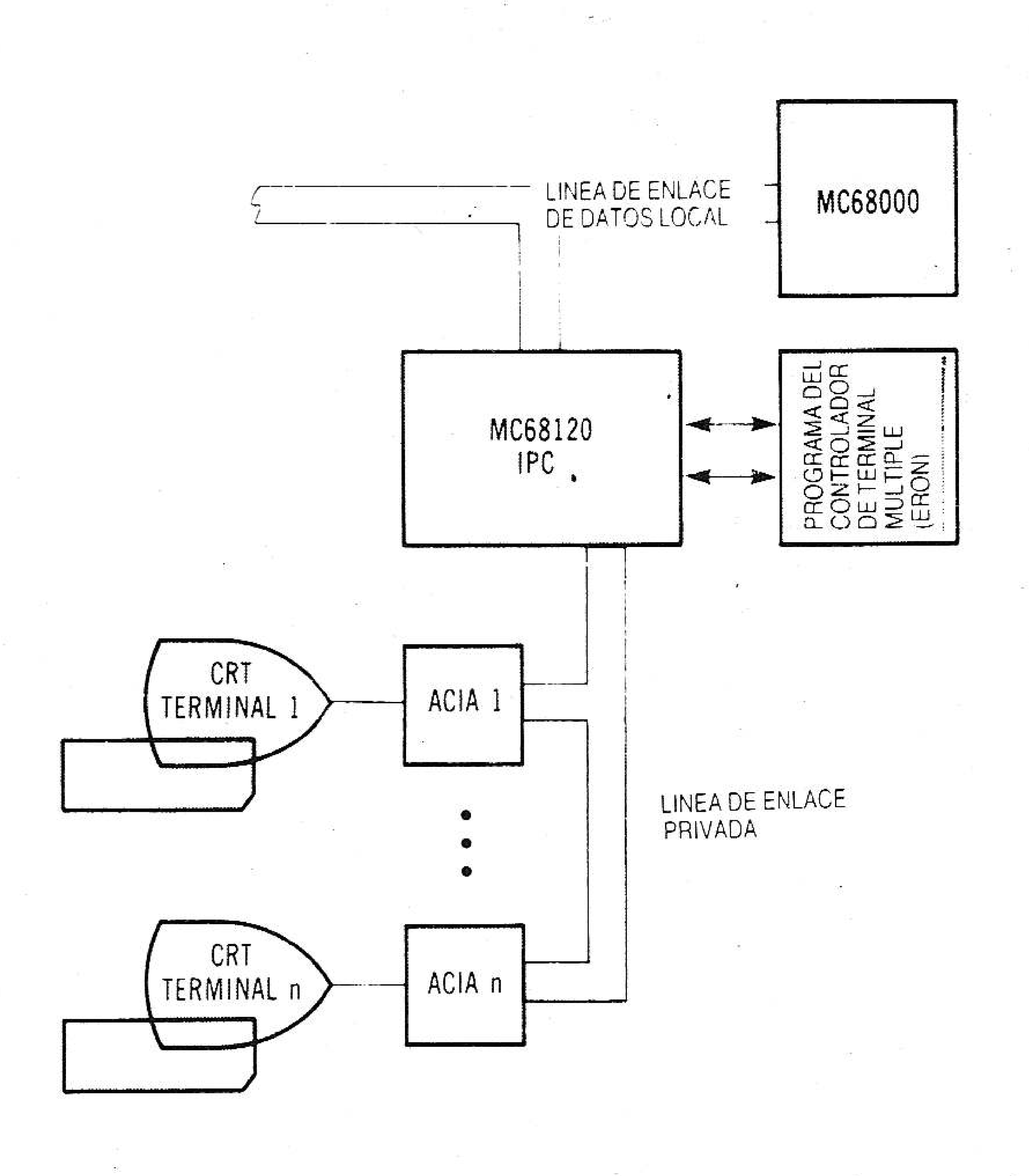


Fig. 4. El controlador periférico inteligente (MC68120) puede supervisar las operaciones del terminal múltiple y reducir enormemente la operación auxiliar de interrupción del procesador central.

Los mensajes que son enviados por la computadora central a los terminales se colocan en un almacenamiento intermedio. El controlador periférico inteligente entonces toma los mensajes y los saca del almacenamiento intermedio y realiza su salida, de una manera seriada, a los puertos seriados de entrada/salida. Los mensajes desde los terminales son almacenados por el controlador periférico inteligente nuevamente de una forma o manera seriada y luego se los envía a la computadora central a través del almacenamiento intermedio del RAM de vías dobles.

Haciendo que el controlador periférico inteligente actúe como un controlador de terminal múltiple reduce el número de interrupciones que la computadora central debe realizar, desde una interrupción por carácter hasta una interrupción por mensaje. Si el mensaje promedio es de una longitud de 20 caracteres, esto disminuye el régimen de solicitud de interrupción por un factor de 19..

La arquitectura del controlador periférico inteligente está relacionada al MC6800 y al microprocesador de un solo chip MC 6801 y se optimiza para las aplicaciones orientadas al control. El procesador de 8 bits es más compatible con tanto sea el código fuente y el código objeto para el MC6800, y muestra el modelo de programación del usuario que se ilustra en la figura 5.

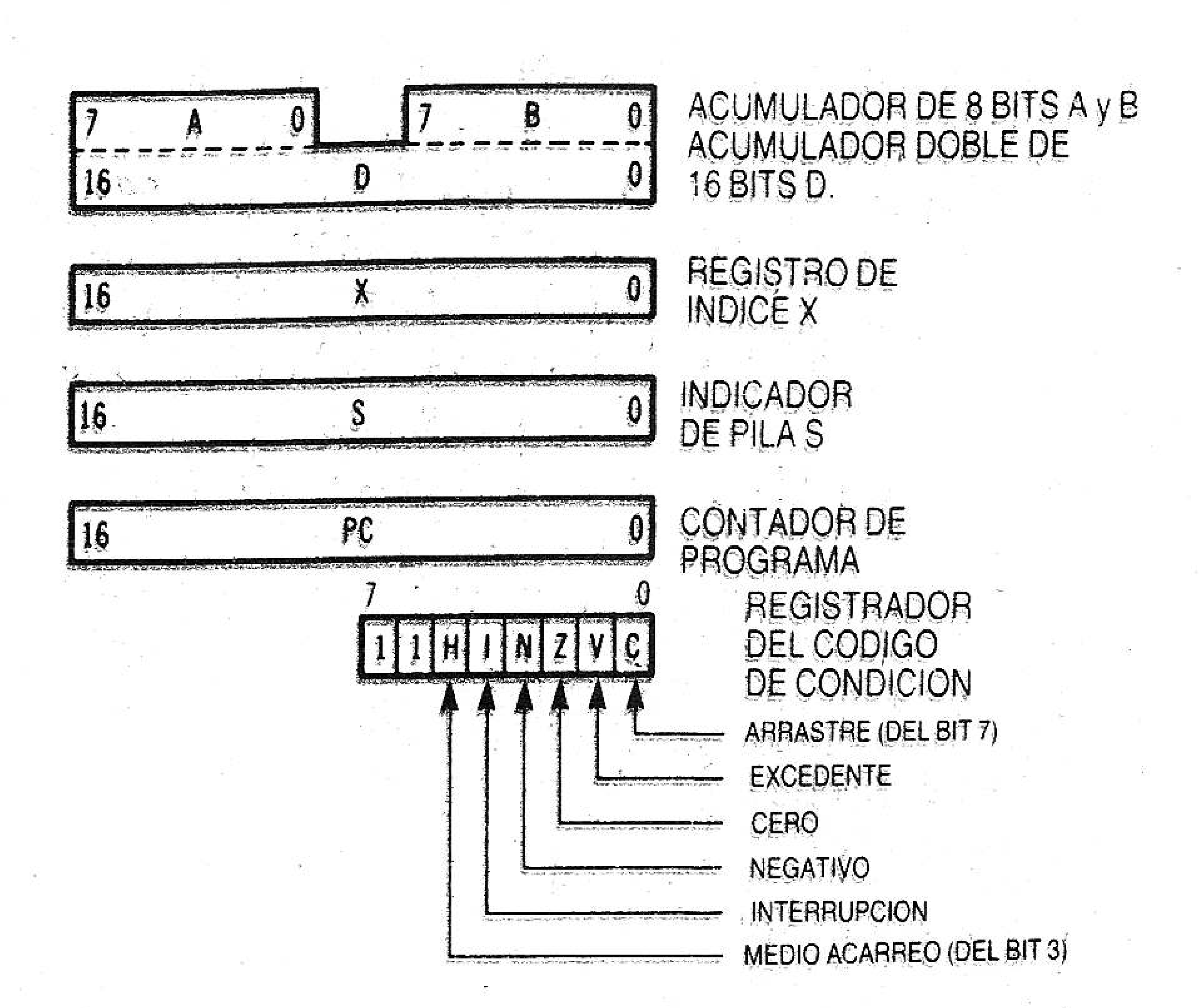


Fig. 5. La programación del usuario del controlador periférico inteligente es directa. Varios tipos de información y esquemas de direccionamiento se soportan con registros semáforo utilizados para control y verificación de estado.

El juego del registrador consiste en dos acumuladores de 8 bits, A y B, que se pueden conectar a un acumulador único de 16 bits, D. Adicionalmente el procesador tiene un registrador de índice de 16 bits, un indicador de pila de 16 bits, y un contador de programa de 16 bits. El registro de código de condición tiene bits para acarreo/acarreo negativo, excedente, cero, negativo, interrupción y medio acarreo.

El juego de instrucción del MC68120 incluye bit de soporte, byte, decimal codificado en binario y tipos de información de palabra de 16 bits. Los modos de direccionamiento incluyen inherente, inmediato, extendido/prolongado, relativo y con índice. El espacio soportado de dirección es 16 bits, correspondiente a 64 Kbytes de memoria. El procesador interno MC68120 se interconecta a la línea de enlace de instrucción del MC68000 a través de un almacenamiento intermedio de RAM de vía doble. Este almacenamiento intermedio de 128 bytes tiene 6 registros semáforos que ayudan al control de su uso. Los registros semáforos contienen bits que indican cuándo una porción específica del almacenamiento intermedio está en uso y qué procesador fue el último en utilizar dicho almacenamiento intermedio. Esto elimina la necesidad de verificar si el controlador periférico inteligente ha leído la RAM de vía doble volviendo a leer el mensaje que queda en el almacenamiento intermedio.

Para los diseñadores que desean un acercamiento más flexible al soporte periférico, el periférico de multifunción MC98901 es una combinación de dispositivos de interface misceláneo que comúnmente se encuentra en los diseños de sistemas pequeños (ver figura 6).

Estos elementos incluyen cuatro temporizadores, un controlador de interrupción que soportará hasta 16 fuentes, una vía de entrada/salida en paralelo de 8 bits, y un receptor/transmisor sincrónico/asincrónico universal de un solo canal para comunicaciones en serie. Los cuatro temporizadores consisten de dos que son prefijables, y dos que son solamente temporizadores de conteo regresivo. Cada temporizador se precalibra individualmente.

El controlador de interrupción puede realizar las prioridades de hasta 16 fuentes de interrupción desde 8 fuentes externas (mediante la vía de entrada/salida en paralelo) (temporizadores internos o vía en serie). La vía en paralelo permite a cada bit ser programado individualmente para que sea una entrada o una salida. Cada pata también puede ser una fuente de interrupción y puede generar una interrupción o una transición de borde entrante/principal o final. La vía en serie receptor/transmisor/sincrónico/asincrónico universal soporta comunicaciones asincrónicas bidireccionales a velocidades de hasta 62,5 kbps. También trabaja en un modo sincrónico donde soportará velocidades de información de hasta 1 M bps. El régimen baudio puede estar generado tanto interna como externamente y el receptor/transmisor sincrónico/asincrónico universal tiene tanto sea modem total y acuses de recibo del controlador de acceso directo a memoria. La línea de enlace de datos de 8 bits del periférico de multifunción se debe colocar en un paquete dual-directo de 48 patas (conexiones).

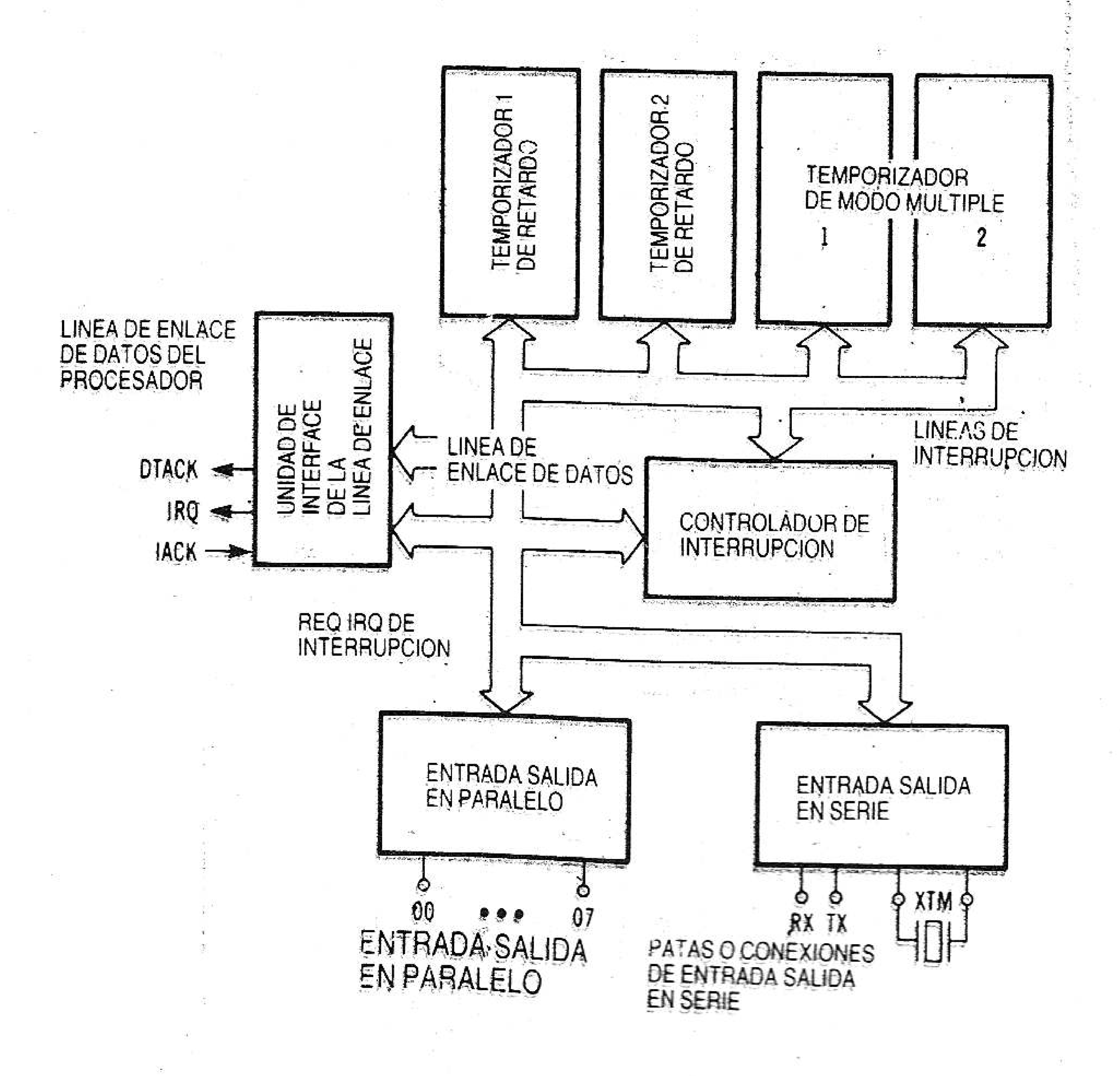


Fig. 6. El chip periférico de multifunción MC68901 puede realizar las prioridades de 16 fuentes de interrupción. Las vías o conexiones en paralelo y en serie permiten que las comunicaciones sincrónicas y asincrónicas se lleven a cabo dentro de este paquete de 48 patas o conexiones.

Interfaces de comunicaciones de datos

Un área de creciente importancia para los fabricantes de sistemas es la comunicación de datos y el sistema de red local. Dado que la mayoría de los sistemas distribuidos se conectan conjuntamente con enlaces de datos de alta velocidad, se pueden encontrar soluciones de integración de gran escala a problemas específicos de circuitos. La tendencia actual en los sistemas de computación de oficina distribuida ubica a la inteligencia en estaciones de trabajo de bajo costo que dependen fuertemente en servidores inteligentes de archivo. Estos sistemas, que deben interconectarse en una forma de acoplamiento suelto utilizan una suerte de forma de comunicación de datos en serie. Con esta tendencia en mente, la línea de periféricos de la familia M68000 ha sido fuertemente orientada hacia las comunicaciones de datos. La tabla 1 compara las velocidades de bit y los protocolos soportados con estas partes que se disponen para soportar estos requerimientos. En esta tabla se encuentran incluidas las partes de la vieja familia M6800 que también trabajan con la MC68000, pero que no se discuten. Luego hay una consideración más aguda a algunos miembros de la familia de comunicación de datos de la M68000.

El MC68562 es un controlador de comunicaciones universal de 2 canales que soporta los protocolos asincrónicos y sincrónicos utilizados comúnmente. Estos protocolos incluyen un procedimiento de control de comunicaciones de datos avanzado, control de enlace de datos sincrónico (SDLC), control mayor de enlace de datos (HDLC) X.25, bisincrónico, y protocolo de mensaje de comunicación de datos digitales. El MC68562 operará a velocidades de información de hasta 4 M bps, y realizará una interface directa a las líneas de información que transporta información de no retorno a cero, no retorno a cero invertido, o codificado Manchester. No se requiere información externa/separación de sincronización, dado que el chip contiene un circuito interbloqueado de fase interna y generadores de regímenes baudio. Tanto los cursos para recibir y transmitir la información están almacenados y separados por cuadruplicado para disminuir la latencia de interrupción para una operación de alta velocidad de información. Además, el chip contiene un temporizador programable que se puede utilizar para la generación de interrupción de impulsos o de funciones de compás de espera o finalización de plazos. La interface de interrupción soporta el esquema de interrupción vectoreado similar al resto de la familia M68000, o las interrupciones autovectoreadas utilizadas en la familia de periféricos M6800. La interface de datos al dispositivo ofrece acuses de recibo de línea de control con modem total y acuses de recibo de acceso directo a memoria para operaciones de movimiento en serie directas a memoria. La interface en paralelo del chip tiene un curso de información en paralelo de 8 bits, y ofrece soporte para los acuses de recibo de la familia M68000. Estas líneas se utilizan para terminación de ciclo, solicitud de interrupción, y reconocimiento de interrupción, respectivamente. Esta parte, actualmente bajo el desarrollo de Signetics, será luego provista por Motorola.

El controlador de red de área local para Ethernet (LANCE) soporta completamente la definición del libro azul de la Norma Ethernet, y ambos mecanismos de transmisión y acceso. Opera con un régimen de datos de 10 M bps, e incluye un controlador de acceso directo a memoria en el lado del procesador de la interface para ayudar a soportar los regímenes de alta información. El controlador de acceso directo a memoria cubre el espacio de dirección total de 24 bits que nos ofrece el MC68000. Además también se soportan las transferencias del acceso directo a memoria en almacenamientos intermedios múltiples de recepción desde los almacenamientos intermedios múltiples de transmisión.

LANCE maneja el arbitraje del acceso múltiple del sentido portador, línea de enlace en serie para detección de colisión, utilizando un algoritmo que incorpora el escuchar antes de hablar, escuchar mientras se habla y la retransmisión sobre la colisión luego de una espera al azar. Al existir colisiones sucesivas, el valor medio de la espera al azar se aumenta para disminuir la probabilidad de otra colisión, y para suministrar un nivelado de carga sobre la línea de enlace en serie. También se incluye en LANCE un soporte para los niveles más bajos del protocolo Ethernet realizando un enmarcado condensado, inserción y extracción de verificación de redundancia de preámbulo y cíclica y un filtrado y detección de direccionamiento. El controlador de acceso directo a memoria maneja todas las transferencias de datos una vez que LANCE ha sido comenzado, de manera tal que las comunicaciones hacia y desde la parte se realizan a través del almacenamiento intermedio de RAM.

Para conservar las conexiones en LANCE, se realiza una multiplexación de las direcciones y de los datos, por lo que se permite que la interface se aloje en un paquete de 48 conexiones.

El controlador inteligente múltiple en disco (IMDC) es un controlador en disco que tiene el propósito de ser usado con mecanismos impulsores Winchester que se disponen corrientemente a costos razonables. Esta parte soporta el tipo de interfaces SA-1000/ST 506, así como los mecanismos impulsores de discos flexibles SA-800. Se puede realizar una interface de cualquier combinación de hasta cuatro mecanismos impulsores de disco con el controlador de una sola vez, permitiendo tanto a los discos rígidos como a los flexibles ser controlados simultáneamente. Esta flexibilidad hace sencillo el respaldo o seguridad (backup) de los discos rígidos por cuanto, para el controlador, el disco flexible aparece como un disco rígido. En tal caso, la operación de respaldo o seguridad se transforma simplemente en una operación de copia sector a sector.

La base de este controlador está construida alrededor de un controlador de acceso directo a memoria. Como resultado, tiene la capacidad de realizar transferencias de archivo hacia y desde los almacenamientos intermedios que se encuentran en ubicaciones arbitrarias en la memoria. El controlador de disco múltiple inteligente está programado para ser colocado en el tercer trimestre de 1983.

Otro chip de soporte orientado en disco que se puede esperar en 1983 es el circuito de traba de fase en dis-

co MC68459. El MC68459 es un circuito de seguridad de fase digital muy rápida que realiza la separación de datos/reloj sobre el flujo de datos entrantes de un disco Winchester. La entrada provista al circuito de traba/seguridad de fase en disco es la señal de datos/reloj compuesta desde el mecanismo impulsor de disco. La salida es una señal separada de datos con una señal de reloj sincronizada. Estas dos señales son importantes al chip del controlador de disco múltiple inteligente por cuanto requiere entradas separadas de datos y reloj. El circuito de traba/seguridad de fase en disco está construído con la tecnología del semiconductor óxido metálico de alto rendimiento y sincronizará y separará un flujo de datos de hasta 20 M bps. Su presentación se programa para el tercer trimestre de 1983.

El controlador dual de acceso directo a memoria es la versión más pequeña y de más bajo costo del controlador de acceso directo a memoria MC68450. Es idéntico al MC68450 con la excepción de su soporte para solamente dos canales de operación. No soporta los modelos de encadenamiento exóticos que realiza el MC68450 tampoco. El MC68440 tiene una compatibilidad directa con el MC68450 pero no tiene señales activas que correspondan a los acuses de recibo del tercer y cuarto canal del acceso directo a memoria. El acceso directo dual a memoria se puede esperar en el cuarto trimestre de 1983.

Expendiendo la familia de chips 68000, tanto hacia arriba o hacia abajo, el alcance y la profundidad de la línea del producto resulta agrandada (ver tabla 2). Esto da como resultado una familia de chips con tales ventajas como un soporte periférico aumentado y de menor costo. Más aún, dado el amplio espectro de sofisticación dentro de la familia, se dispone de elecciones alternativas para el diseñador.



Tabla 2 - Familia del 68000

Dispositivo	Función	Disponibilidad
68000	MPU de 16 bits	Inmediata
68008	68000 de 8 bits	Inmediata*
68010	procesador VM	Inmediata*
68020	68000 de 32 bits	4to. trimestre 1983
68120	IPC	Inmediata
68121	IPC - NR	Inmediata
68200	MCU	2do. trimestre 1983
68230	PI/T	Inmediata
68430	DMAI	4to. trimestre 1983
68440	DDMA	4to. trimestre 1983
68450	DMAC	Inmediata
68451	MMU	Inmediata
68452	BAM	Inmediata
68454	IMDC	3er. trimestre 1983
68459	DPLL	3er. trimestre 1983
68561	MPCC - II	Inmediata*
68562	DUSCC	3er. trimestre 1983
68564	SIO	Inmediata*
68590	LANCE	3er. trimestre 1983
68652	MPCC	Inmediata
68653	PGC	INmediata
68661	EPCI	Inmediata
68681	DUART	Inmediata
68881	FPCP	4to. trimestre 1983
68901	MFP	Inmediata*

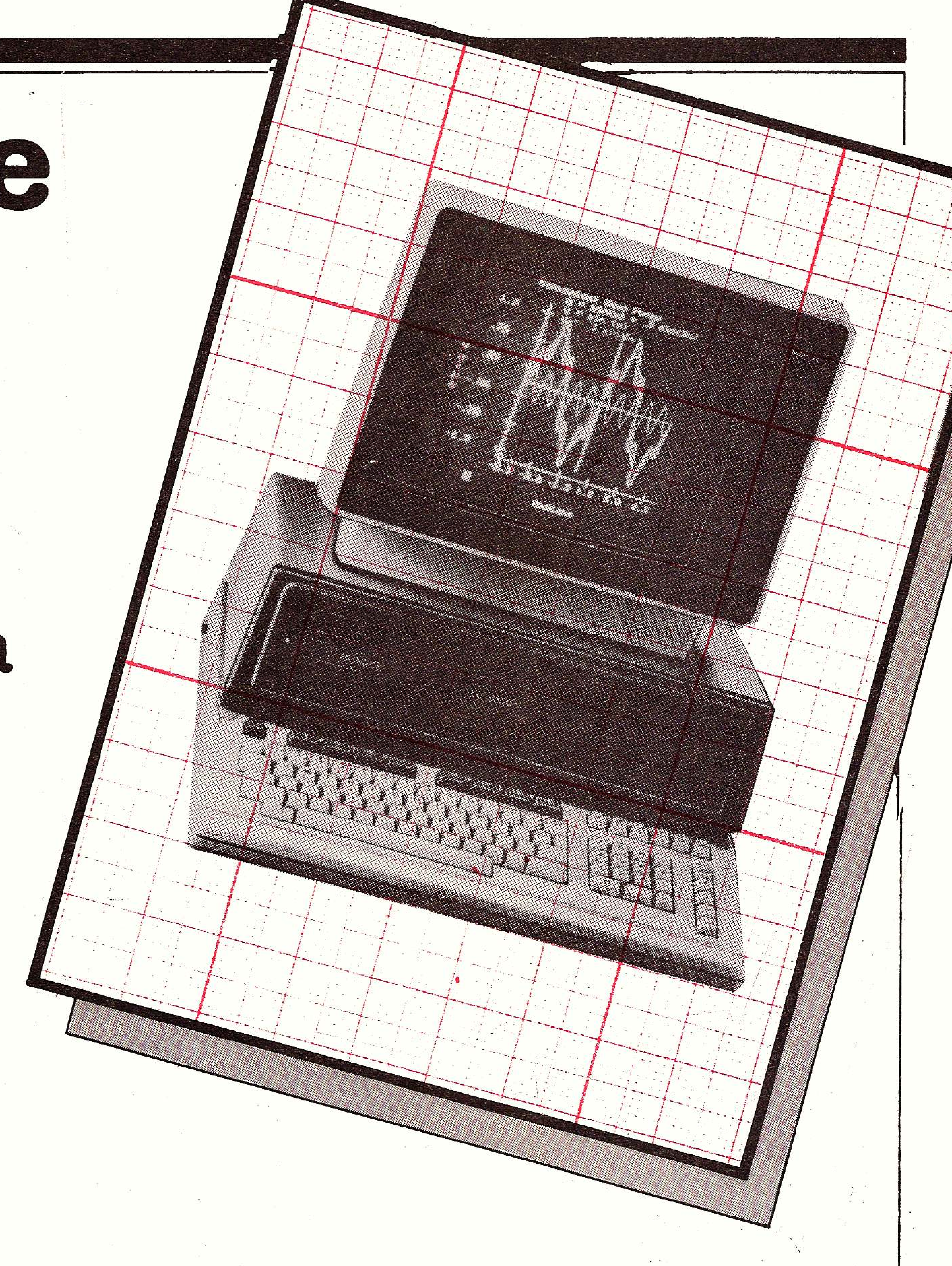
*Dispositivos ya disponibles en el 4to. trimestre de 1982, pueden ser aún cantidades de muestras.

Tabla 1 - Dispositivos de comunicación de datos M68000

Dispositivo	Protocolo	Velocidad
MC68050 ACIA	asincrónico	0 a 1 M bps
MC68562 MPCC		
MC68564 SIO		
MC68661 EPCI		
MC68681 DUART		
MC68052 SSDA	bisincrónico, orientado en bit	
MC68562 MPCC		
MC68564 SIO		
MC68652 MPCC		
MC68661 EPCI		
MC68564 ADLC	SDLC/HDLC, protocolo de control de byte.	
MC68562 DUSCC		
MC68564 SIO		
MC68652 MPCC		
MC68052 SSDA	protocolo de control de byte	1 a 2 M bps
MC68562 DUSCC		
MC68652 MPCC		
MC68054 ADLC	protocolo orientado en bit	
MC68562 DUSCC		
MC68652 MPCC		
MC68562 DUSCC	protocolo de control de byte	2 a 4 M bps
MC68562 DUSCC	protocolo orientado en bit	

Sigla	Definición
MPU	Unidad de microprocesamiento
VM	Máquina virtual
IPC	Controlador periférico inteligente
NR	Sin ROM
MCU	Unidad de micro control
PI/T	Interface/temporizador en paralelo
DMAI	Interface de acceso directo a memoria
DDMA	Controlador de acceso directo a memoria dual
MMU	Unidad de gestión de memoria
BAM	Módulo de arbitraje de línea de enlace
IMDC	Controlador de disco múltiple inteligente
DPLL	Circuito de traba/seguridad de fase en disco
MPCC	Controlador de comunicaciones del protocolo múltiple
DUSCC	Controlador de comunicaciones unviersal de canal dual/doble
SIO	Entrada/salida en serie
LANCE	Controlador de red de área local para Ethernet
PGC	Generador y verificador polinomial
EPCI	Interface de comunicaciones programable incrementado
DUART	Receptor/trasmisor asincrónico universal dual
FPCP	Coprocesador de punto flotante
MFP	Periférico de función múltiple

Monroe EC 8800



La computadora educativa. Un nuevo acceso en el mercado de la educación

El uso de computadoras en la educación es en la actualidad variado. La computadora se utiliza para enseñar programación y desarrollar la familiaridad con los lenguajes; como máquina de enseñanza, con software diseñado para enseñar partes específicas del plan de estudios; y además como una herramienta administrativa.

Monroe Systems for business anunció recientemente su acceso al mercado de la computación educativa con la introducción de la EC 8800.

El objetivo de Monroe con la EC 8800 apuntaba a cubrir las necesidades aún no resueltas.

Video

La EC 8800 utiliza un monitor de color RGB o un monitor de video compuesto monocromático como exhibidor. Elimina la necesidad del circuito de video compuesto y de un modulador RF.

Disco

La EC 8800 utiliza un mecanismo impulsor de disco de 5 1/4", incorporado en la caja y ya viene con la computadora. Una característica inusual del sistema de disco es su capacidad para almacenar datos: 320K en un disco, con un formato de 16 sectores y 80 pistas por disco (doble densidad).

Además, hay un segundo mecanismo impulsor de disco disponible que puede aumentar la capacidad de almacenamiento de un sistema a 640K. Dado que el acceso a la información es rápido y la capacidad de almacenamiento es rápida, las aplicaciones que normalmente requieren que los discos se cambien en la mitad del programa o que se agreguen mecanismos impulsores extras puede frecuentemente ser realizado con un disco y un mecanismo impulsor.

Memoria

La pantalla de texto tiene caracteres en minúscula y mayúscula así como caracteres de doble tamaño e intermitentes. Además hay gráficos de baja y alta resolución. En la

pantalla de baja resolución se puede utilizar tanto sea bloques individuales (un reticulado de 72 x 80) o caracteres gráficos de baja resolución (un reticulado de 20 x 40). Los caracteres gráficos de baja resolución son del msmo tamaño que un caracter textp, similar a los caracteres gráficos de Pet, pero en color. Cada letra representa una configuración diferente de 6 bloques, y cuando se la imprime luego de una orden de baja resolución, se producirá una combinación de estos bloques. Estos se pueden combinar para producir formas de baja resolución que sean sencillas para desplazar o mover dentro de la pantalla.

En el modo de baja resolución se dispone de 8 colores que se pueden utilizar en su totalidad al mismo tiempo.

Las órdenes Basic le permiten al usuario graficar puntos y líneas entre puntos, pintar objetos de forma iregular, completar rectángulos, crear círculos, definir una parte de la pantalla como una serie, y realizar tablas. La parte de la pantalla que se define como una serie puede entonces imprimirse como tal en cualquier ubicación de la pantalla. Las tablas que se crean de esta manera se pueden rotar, representar en escalas y borrarse en cualquier punto de la pantalla. Hay también un modo de animación con dos colores que hace posible una animación rápida.

Sonido

La EC 8800 contiene un panel standard de sonido que produce 4 canales programables. La respuesta de frecuencia es de 92 a 20.000 Hz. Se suministra un control de volumen programable así como manual. El parlante interno es standard así como lo es la salida de 8 ohms en el panel posterior.

Vías de Acceso

Hay varias vías de acceso standard en la EC 8800: 3 vías RS-232 y un conector de 50 patas para una línea de distribución interna. Esto hace relativamente simple la conexión de periféricos. Las vías de acceso RS-232 tiene regímenes baudio programables y protocolos. Uno es una vía para comunicaciones, y los otros dos son para la conexión de periféricos. Existe también un toma para corriente alterna y una conexión de salida de audio para auriculares, así como dos conectores para palanca de mandos u otros dispositivos de comando para juegos.

Periféricos y Planes Futuros

Actualmente Monroe ofrece una tableta de gráficos y un modem acústico como periférico además de 5 impresoras.

Los planes futuros incluyen un disco rígido y un controlador de grupos que le permitirán a la EC 8800 comunicarse una con otra y compartir los recursos comunes de los periféricos tales como los discos rígidos y las impresoras.

Las impresoras por matriz de puntos tienen la capacidad (en PROM) (memoria sólo para lectura, programable) de manejar el copiado de texto y gráficos de la pantalla y de los archivos en disco para la impresora en blanco y negro. Esto incluye la capacidad de imprimir los caracteres gráficos de baja resolución mencionados anteriormente.

Software

El software para la computadora incluye un sistema operativo multitarea, manejo de archivos ISAM, 32K Basic, Monroe Pascal y Pilot. También se dispone de CP/M.



SCANALI

Este programa, produce un balance al final de cada período computado. Se requiere un ingreso diario que actualiza cuentas de capital para ganancia o pérdida neta, para asegurar un equilibrio de capital propio, pasivo y cuentas de capital.

```
10 CLEAR 900
20 REM           SCANALI
30 CLS
40 N1=15
50 N5=5
60 DIM R$(N1),V$(N1),S$(N1)
70 DIM R(N1),V(N1)
80 DIM T$(5),N0(5)
90 PRINT "INGRESE EL NOMBRE DEL ARCHIVO DE CUENTAS";
100 INPUT F$
110 GOSUB 320
120 K=1
130 GOSUB 390
140 X=CVI(R$(1))
150 N2=CVI(R$(2))
160 N3=X
170 PRINT
180 PRINT "LA FECHA DE LA ULTIMA ACTUALIZACION FUE :";D$
190 FOR K=2 TO N2+1
200 GOSUB 390
210 N0(K-1)=CVI(R$(1))
220 N3=N3+N0(K-1)
230 T$(K-1)=D$
240 NEXT K
250 PRINT
260 GOSUB 410
270 PRINT
280 PRINT
290 PRINT "PROCESAMIENTO COMPLETO"
300 PRINT
310 STOP
320 OPEN "R",1,F$
330 FIELD 1,19 AS D$
340 FOR I= 1 TO N1
350 FIELD 1,19+(I+1)*7 AS Z$,2 AS R$(I),1 AS S$(I),4 AS V$(I)
360 NEXT I
370 FIELD 1,124 AS X$,2 AS L$,2 AS N$
380 RETURN
390 GET 1,K
400 RETURN
410 PRINT "INGRESE LA FECHA DE HOY ";
420 INPUT D4$
430 PRINT
440 PRINT "PREPARE LA IMPRESORA - DIGITE <ENTER> CUANDO ESTE LISTA";
450 INPUT A$
460 LPRINT" "
470 LPRINT TAB(30);F$
480 LPRINT TAB(30);"BALANCE"
490 LPRINT TAB(30);D4$
500 LPRINT" "
510 LPRINT" "
520 T=50
530 FOR J=1 TO 3
540 IF J=1 THEN LPRINT TAB(30);T$(1)
550 IF J<>2 THEN 590
560 T=40
570 LPRINT TAB(25);"DISPONIBILIDADES Y CAPITAL"
580 A2=0
590 LPRINT" "
600 LPRINT TAB(5);T$(J)
610 K1=X+1+N7
620 N7=N7+N0(J)
630 FOR I=K1 TO K1+N0(J)-1
640 K=I
650 GOSUB 390
660 IF K>N3 THEN LSET D$=D2$
670 FOR J1=1 TO N1
680 R1=CVI(R$(J1))
690 IF R1=0 THEN 740
700 V(0)=CVS(V$(J1))
710 IF S$(J1)="C" THEN A0=A0-V(0)
720 IF S$(J1)="D" THEN A0=A0+V(0)
730 NEXT J1
740 N=CVI(N$)
750 IF N<=0 THEN 790
760 D2$=D$
770 K=N
780 GOTO 650
790 IF J>1 THEN A0=A0*(-1)
800 LPRINT TAB(10);I;"- ";D$;TAB(T);A0
810 A1=A1+A0
820 A0=0
830 NEXT I
840 LPRINT TAB(T-2);"------------------"
850 LPRINT  TAB(5),"TOTAL ";T$(J);;TAB(50);A1
860 IF J=1 THEN LPRINT TAB(48);"=================="
870 LPRINT" "
880 A2=A2+A1
890 A1=0
900 NEXT J
910 LPRINT TAB(48);"------------------"
920 LPRINT TAB(5);"TOTAL DE DISPONIBILIDADES  Y CAPITAL";TAB(50);A2
930 LPRINT TAB(48);"=================="
1000 RETURN
```

SCCOMPUT

Este programa produce una comparación de egresos (opción E) o de ingresos (opción I) de varios períodos previos. El producto total formateado incluye datos detallados, y el promedio para cada cuenta y período. Observe que los registros con el código de transacción "*" son usados como ingreso a este programa. El nro. de períodos a comparar es especificado en respuesta al programa listo.

```
10 CLEAR 900
20 REM               SCCOMPUT
30 CLS
40 N1=15
50 T=10
60 N5=5
70 DIM R$(N1),V$(N1),S$(N1)
80 DIM R(N1),V(N1)
90 DIM T$(5),N0(5)
100 PRINT "INGRESE EL NOMBRE DEL ARCHIVO DE CUENTAS";
110 INPUT F$
120 GOSUB 330
130 K=1
140 GOSUB 400
150 X=CVI(R$(1))
160 N2=CVI(R$(2))
170 N3=X
180 PRINT
190 PRINT "LA FECHA DE LA ULTIMA ACTUALIZACION FUE ";D$
200 FOR K=2 TO N2+1
210 GOSUB 400
220 N0(K-1)=CVI(R$(1))
230 N3=N3+N0(K-1)
240 T$(K-1)=D$
250 NEXT K
260 PRINT
270 GOSUB 420
280 PRINT
290 PRINT
300 PRINT "PROCESAMIENTO COMPLETO"
310 PRINT
320 STOP
330 OPEN "R",1,F$
340 FIELD 1,19 AS D$
350 FOR I= 1 TO N1
360 FIELD 1,19+(I-1)*7 AS X$,2 AS R$(I),1 AS S$(I),4 AS V$(I)
370 NEXT I
380 FIELD 1,124 AS X$,2 AS L$,2 AS N$
390 RETURN
400 GET 1,K
410 RETURN
420 PRINT "CUANTOS SE INCLUYEN"
430 INPUT N9
440 PRINT "DESEA COMPARAR INGRESOS O EGRESOS <I> <E>"
450 INPUT A$
460 J=4
470 IF A$="E" THEN J=5
480 DIM A2(N9+1)
490 PRINT
500 PRINT "PREPARE LA IMPRESORA - DIGITE <ENTER> CUANDO ESTE LISTA";
510 INPUT A$
520 LPRINT" "
530 LPRINT TAB(30);F$
540 LPRINT TAB(25);"COMPARACION DE ";T$(J)
550 LPRINT TAB(30);D4$
560 LPRINT" "
570 FOR I1=1 TO N9
580 LPRINT TAB(T*(I1-1)+20);"PER.";I1;
590 NEXT I1
600 LPRINT "                 TOTAL              PROMEDIO"
610 LPRINT" "
620 N7=N0(1)+N0(2)+N0(3)
630 IF J=5 THEN N7=N7+N0(4)
640 K1=X+1+N7
650 N7=N7+N0(J)
660 FOR I=K1 TO K1+N0(J)-1
670 I1=N9
680 K=I
690 GOSUB 400
700 IF K>N3 THEN LSET D$=D2$
710 FOR J1=N1 TO 1 STEP -1
720 V(0)=CVS(V$(J1))
730 IF S$(J1)<>"*" THEN 780
740 V(I1)=V(0)
750 IF J=4 THEN  V(I1)=V(I1)*(-1)
760 I1=I1-1
770 IF I1=0 THEN 790
780 NEXT J1
790 LPRINT D$;
800 FOR I1=1 TO N9+1
810 IF I1<=N9 THEN V(N9+1)=V(N9+1)+V(I1)
820 A2(I1)=A2(I1)+V(I1)
830 LPRINT TAB(T*(I1-1)+20);V(I1);
840 NEXT I1
850 LPRINT TAB(T*(I1-1)+20);V(N9+1)/N9
860 V(N9+1)=0
870 NEXT I
880 LPRINT " "
890 LPRINT TAB(5);"TOTAL ";
900 FOR I1=1 TO N9+1
910 LPRINT TAB(T*(I1-1)+20);A2(I1);
920 NEXT I1
930 LPRINT TAB(T*(I1-1)+20);A2(I1-1)/N9
940 LPRINT" "
950 RETURN
```

```
10 CLEAR 900
20 '                SCDIARIO
30 CLS
40 N1=15
50 M1=999999
60 M2=0
70 N5=5
80 DIM R$(N1),V$(N1),S$(N1)
90 DIM R(N1),V(N1)
100 DIM T$(5),N0(5)
110 PRINT "INGRESE EL NOMBRE DEL ARCHIVO DE CUENTAS";
120 INPUT F$
130 GOSUB 460
140 K=1
150 GOSUB 530
160 X=CVI(R$(1))
170 N2=CVI(R$(2))
180 N3=X
190 PRINT
200 PRINT "LA FECHA DE LA ULTIMA ACTUALIZACION FUE ";D$
210 FOR K=2 TO N2+1
220 GOSUB 530
230 N0(K-1)=CVI(R$(1))
240 N3=N3+N0(K-1)
250 T$(K-1)=D$
260 NEXT K
270 PRINT
280 PRINT "DIGITE UNA OPCION :"
290 PRINT
300 PRINT TAB(5);1;".....COMIENZO Y FINAL DE NUMEROS DE REFERENCIA"
310 PRINT TAB(5);2;".....ENTRADAS DIARIAS ORDENADAS"
320 PRINT
330 PRINT "INGRESE UNA OPCION";
340 INPUT O
350 PRINT
360 PRINT
370 IF O-1 THEN GOSUB 550
380 IF O<>2 THEN 420
390 GOSUB 550
400 GOSUB 700
410 PRINT
420 PRINT
430 PRINT "PROCESAMIENTO COMPLETO"
440 PRINT
450 STOP
460 OPEN "R",1,F$
470 FIELD 1,19 AS D$
480 FOR I=1 TO N1
490 FIELD 1,19+(I-1)*7 AS X$,2 AS R$(I),1 AS S$(I),4 AS V$(I)
500 NEXT I
510 FIELD 1,124 AS X$,2 AS L$,2 AS N$
520 RETURN
530 GET 1,K
540 RETURN
550 FOR K=X+1 TO N3
560 GOSUB 530
570 FOR I=1 TO N1
580 R1=CVI(R$(I))
590 IF R1<=1 THEN 620
600 IF R1<M1 THEN M1=R1
610 IF R1>M2 THEN M2=R1
620 NEXT I
```

SCDIARIO

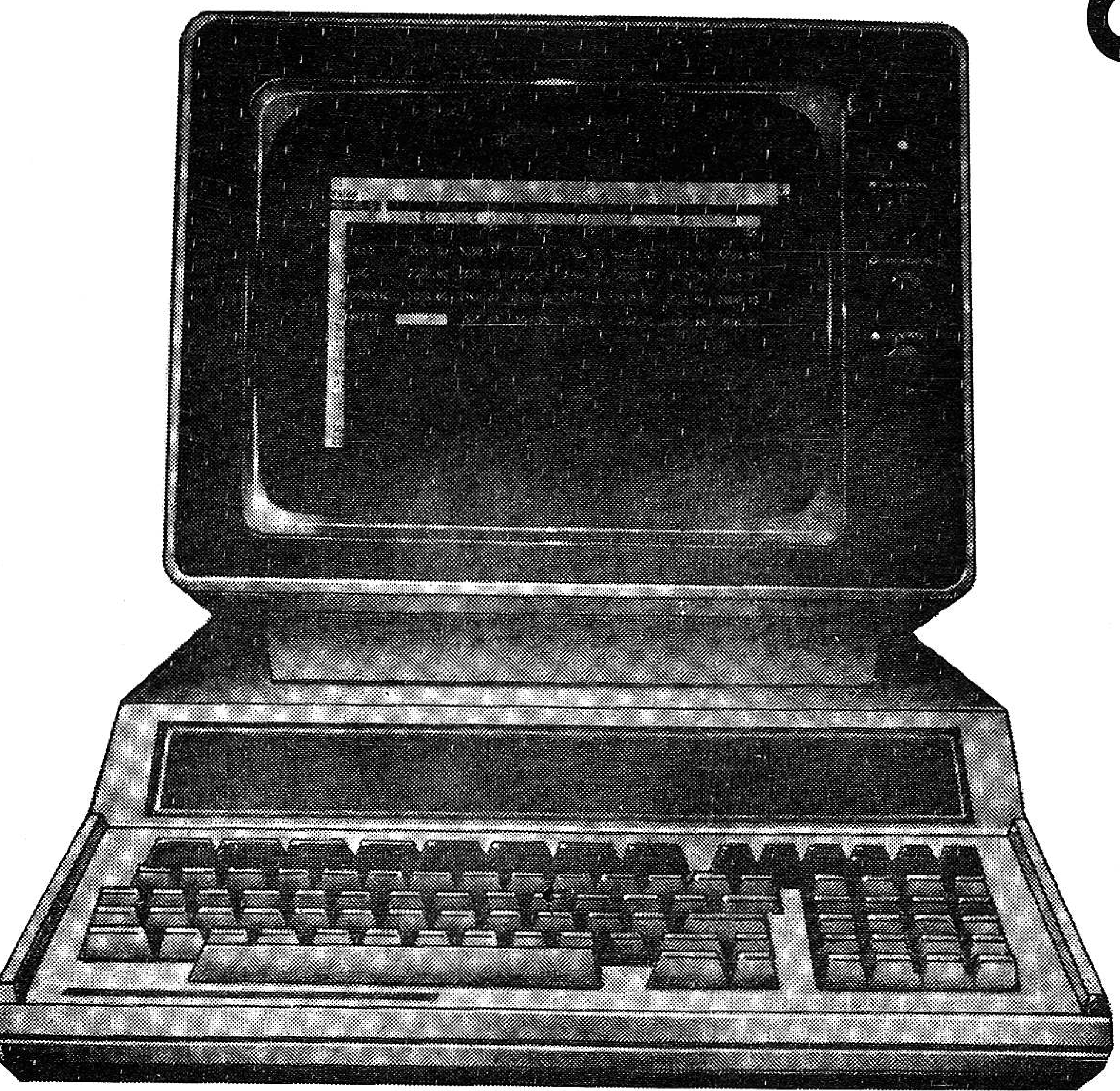
Este programa vuelve a crear entradas diarias que fueron ingresadas en el archivo de cuentas. Las entradas impresas, son formateadas de tal modo que facilita la comparación con el diario actual.

```
630 N=CVI(N$)
640 IF N<=0 THEN 670
650 GET 1,N
660 GOTO 570
670 NEXT K
680 PRINT "REFERENCIAS DIARIAS FUERON INGRESADAS ";M1;"A";M2;"INCLUSIVE."
690 RETURN
700 N4=M2-M1+1
710 N6=M1-1
720 DIM E(N4,N5)
730 FOR K=X TO N3
740 GOSUB 530
750 FOR I=1 TO N1
760 R1=CVI(R$(I))
770 IF R1=0 THEN 840
780 J1=R1-N6
790 FOR J2=1 TO N5
800 IF E(J1,J2)<>0 THEN 830
810 E(J1,J2)=K
820 GOTO 840
830 NEXT J2
840 NEXT I
850 N=CVI(N$)
860 IF N<=0 THEN 890
870 GET 1,N
880 GOTO 750
890 NEXT K
900 PRINT
910 PRINT
920 PRINT TAB(20);"LISTANDO REFERENCIAS DIARIAS"
930 PRINT
940 PRINT
950 PRINT "REF";TAB(15);"CUENTA";TAB(40);"DEBITO";TAB(50);"CREDITO"
960 PRINT
970 FOR J1=1 TO N4
980 PRINT J1+N6;
990 FOR J2=1 TO N5
1000 IF E(J1,J2)=0 THEN 1200
1010 K=E(J1,J2)
1020 GOSUB 530
1030 IF K<=N3 THEN 1060
1040 K=CVI(L$)
1050 LSET D$=D2$
1060 FOR I=1 TO N1
1070 R1=CVI(R$(I))
1080 IF R1=0 THEN 1160
1090 IF R1<>J1+N6 THEN 1160
1100 V(0)=CVS(V$(I))
1110 T=0
1120 IF S$(I)="C" THEN T=10
1130 PRINT
1140 PRINT TAB(10+T);K;TAB(15+T);"- ";D$;TAB(40+T);V(0)
1150 GOTO 1230
1160 NEXT I
1170 D2$=D$
1180 K=CVI(N$)
1190 GOTO 1020
1200 IF E(J1,1)<>0 THEN 1230
1210 PRINT "***** NO GRABADO *****"
1220 GOTO 1250
1230 NEXT J2
1240 PRINT
1250 NEXT J1
1260 RETURN
```

Hay dos opciones para su uso:

OPCION I. produce un reporte indicando el comienzo y fin de los números de referencia que han sido ingresados diariamente.

OPCION II. produce la salida de la opción I más el listado de referencia diaria.




La empresa
norteamericana
líder en el mundo
de las
Microcomputadoras



Mallcon & Rockwell

PASCAL PARA EL TRS-80

Como conclusión del número anterior en el cual se marcaron algunas de las diferencias entre los lenguajes PASCAL y BASIC, publicamos a continuación dos ejemplos de programas extraídos de la revista Creative Computing. En los mismos se observará la simplicidad de uno con respecto a otro.

```
PROGRAMA REVERSE;
VAR LEN, LOOP1, LOOP2: INTEGER;
  TARGET, NEWTARGET: STRING;
  CH: CHAR;
BEGIN
  WRITELN('ENTER YOUR TARGET STRING.');
  READLN(TARGET);
  NEWTARGET:=TARGET;
  LEN:=LENGTH(TARGET);
  FOR LOOP1:=1 TO 100 DO
    BEGIN
      FOR LOOP2:=LEN DOWNTO 1 DO
        BEGIN
        CH:=TARGET[LOOP2 ;
        NEWTARGET[LEN-LOOP2+1: :=CH;
        END; (*LOOP2*)
      TARGET:=NEWTARGET;
      WRITELN(TARGET)
    END (*LOOP1*)
END. (*REVERSE*)
```

```
10 CLEAR 500
20 DEFSTR T
30 PRINT "ENTER YOUR TARGET STRING."
40 LINEINPUT TARGT
50 FOR I1%=1 TO 100
60 T2ARGT=""
70 FOR I2%=LEN(TARGT) TO 1 STEP-1
80 T2ARGT=T2ARGT+MID$(TARG,I2%,1)
90 NEXT I2%
100 TARGT=T2ARGT
110 PRINT TARGT
120 NEXT I1%
```

```
PROGRAMA FACTROOT
CONST EPSILON=1E-5;
VAR LOWLIMIT, HIGHLIMIT, LOOPCOUNT: 1..20;
  HOLD: REAL;
FUNCTION FACTORIAL (VALUE: INTEGER): REAL;
VAR LOOPCOUNT: 1..20;
  PRODUCT: REAL;
BEGIN
  PRODUCT:=1;
  FOR LOOPCOUNT:= VALUE DOWNTO 1 DO
    PRODUCT:=PRODUCT*LOOPCOUNT;
  FACTORIAL:=PRODUCT
END; (*FACTORIAL*)
FUNCTION NEWTON (START: REAL): REAL;
VAR SQROOT: REAL;
BEGIN
  SQROOT:=1;
  REPEAT
    SQROOT:=(SQROOT+START/SQROOT)/2
  UNTIL ABS(START/SQR(SQROOT)-1)<=EPSILON;
  NEWTON:=SQROOT
END; (*NEWTON*)
BEGIN
  WRITELN('FACTORIAL AND ROOT COMPUTATION.');
  WRITELN; WRITE('ENTER LOW LIMIT (0<LOW<21)    ==> ');
  READLN (LOWLIMIT);
  WRITE ('ENTER HIGH LIMIT (1<HIGH<21) ==> ');
  READLN(HIGHLIMIT);
  FOR LOOPCOUNT:=LOWLIMIT TO HIGHLIMIT DO
    BEGIN
      HOLD:=FACTORIAL(LOOPCOUNT);
      WRITELN(LOOPCOUNT:10, HOLD:10, NEWTON(HOLD):10)
    END;
END. (*FACTROOT*)
```

```
10 DEFINT H,L
20 EP=1E-5
30 P1$="###########": P2$="      ##.#####[[[["
40 CLS: PRINT "FACTORIAL AND ROOT COMPUTATION."
50 PRINT: LINEINPUT "ENTER LOW LIMIT (0<LOW<21)  ==> "; LOWLIMIT$
60 LOWLIMIT=VAL(LOWLIMIT$)
70 IF LOWLIMIT<1 OR LOWLIMIT>20 THEN PRINT "ERROR.":END
80 PRINT: LINEINPUT "ENTER HIGH LIMIT (1<HIGH<21) ==> "; HIGHLIMIT$
90 HIGHLIMIT=VAL(HIGHLIMIT$)
100 IF HIGHLIMIT<1 OR HIGHLIMIT>20 THEN PRINT "ERROR.":END
110 FOR LOOPCOUNT=LOWLIMIT TO HIGHLIMIT
120 GOSUB 900
130 GOSUB 1000
140 PRINT USING P1$; LOWLIMIT;: PRINT USING P2$; RESULT; ROOT
150 NEXT
160 END
900 RESULT=1
910 FOR I%=LOOPCOUNT TO 1 STEP-1
920 RESULT=RESULT*I%
930 NEXT
940 RETURN
1000 ROOT=1
1010 ROOT=(ROOT+RESULT/ROOT)/2
1020 IF ABS(RESULT/(ROOT[2)-1)>EP THEN 1010
1030 RETURN
```

Planeocalc (Ultima Parte)


```
404 IFY3<Y4THEN402ELSERETURN
405 IFIN$="N"THEN401ELSERETURN
406 IFY4<1THENY4=1:RETURNELSEY4=IN:RETURN
407 IFF=2THEN402ELSERETURN
408 CLS:GG$="COLUMNA":Y2=CM:GOSUB547:P4=CM:GOSUB548:Y1=IN:PL=576:GOSUB537:IFF=2THEN408ELSEFORX=1TORW:A(Y1,X)=0:NEXT:CLS:GOTO308
409 CLS:GG$="RENGLON":Y2=RW:GOSUB547:P4=CM:GOSUB548:Y1=IN:PL=576:GOSUB537:FORX=1TOCM:A(X,Y1)=0:NEXT:CLS:GOTO308
410 CLS:PRINT@26,"CAMBIA COLUMNAS";:PRINT@512,"QUE COLUMNA DESEA MOVER ";:FL=-2:P4=CM:GOSUB548:Y1=IN:PRINT" DESTINO DE LA COLUMNA";:
GOSUB548:Y2=IN:FORX=1TORW:J1=A(Y1,X):J2=A(Y2,X):A(Y1,X)=J2:A(Y2,X)=J1:NEXTX:CQ$=CM$(Y1):CM$(Y1)=CM$(Y2)
411 CM$(Y2)=CQ$:CQ$="":GOTO308
412 CLS:PRINT@24,"CAMBIO DE RENGLONES":PRINT@512,"RENGLON A CAMBIAR ";:FL=-2:P4=RW:GOSUB548:PRINTTAB(5);:Y1=IN:PRINT" DESTINO DEL RE
NGLON ";:GOSUB548:IFF=2THEN412ELSEY2=IN:FORX=1TOCM:J1=A(X,Y1):J2=A(X,Y2):A(X,Y1)=J2:A(X,Y2)=J1:NEXTX:CQ$=RW$(Y1)
413 RW$(Y1)=RW$(Y2):RW$(Y2)=CQ$:CQ$="":GOTO308
414 PRINT@960,"DESEA INPRIMIR ESTA PANTALLA (S/N)";:FL=2:RETURN
415 CLS:GOSUB416:GOTO417
416 FORX=1TOCM:FORX1=1TO13:FU(X,X1)=0:NEXTX1,X:RETURN
417 CLS:PRINT@23,"INGRESE CALCULOS";:PRINT@704,"";:P5=51:CH=191:GOSUB523:Y1=754:FORX=1TO4:Y1=Y1+64:PRINT@Y1,CHR$(191);:NEXT:PRINT@75
6,"FUNCIONES";:PRINT@819,"1 +";:PRINT@883,"2 -";:PRINT@947,"3 X";:PRINT@1011,"4 /";:PRINT@824,"5 MENU";
418 PRINT@888,"6 PROXIMO";:PRINT@954,"COLUMNA";:Z2=0:FORX=1TO10:Z2=Z2+64:FORSA=1TO7:ONSAGOSUB424,525,425,525,426,525,424:NEXTSA:P4=C
M:P7=Z3:GOSUB548:Z3=P7:C1=IN:PRINT@Z2,"COL.";:PRINTUSING"##";C1;:PRINTCHR$(191);:GOSUB428
419 GOSUB426:P7=Z3:GOSUB548:Z3=P7:IFIN=C1THEN419ELSEFU(C1,13)=IN:PRINT@Z3,"COL";IN;:FORX1=1TO12STEP2:GOSUB428:GOSUB425:P4=6:P7=Z3:GO
SUB548:Z3=P7:PRINT@Z3,"";:FU(C1,X1)=IN:IFIN=1PRINT"+";ELSEIFIN=2PRINT"-";ELSEIFIN=3PRINT"X";ELSEIFIN=4PRINT"/";
420 IFIN=5THEN422ELSEIFIN=6PRINT"FIN";:X1=14:NEXTX1,X:GOTO421ELSEGOSUB428:P4=CM:GOSUB426:P7=Z3:GOSUB548:Z3=P7:FU(C1,X1+1)=IN:PRINT@Z
3,"COL.";IN;:NEXTX1,X
421 PRINT@960,"DESEA IMPRIMIR ESTA PANTALLA (S/N)";:FL=1:GOSUB515:IFIN$="S"THENGOSUB518:GOTO417ELSE417
422 CLS:PRINT@512,"DESEA LA IMPRESION DE LOS CALCULOS (S/N) ";:FL=1:GOSUB515:IFIN$="S"THENGOSUB344
423 CLS:PRINT@512," 1.REGRESA A MENU  O  2. PERFORMANCE DE CALCULOS ";:FL=-1:GOSUB515:IFIN=1THEN308ELSEIFIN=2THEN485ELSE423
424 PRINT@768,"INGRESE LA COL.PARA PERFORMANCE DE CALCULOS ";:FL=-2:RETURN
425 PRINT@833,"FUNCION ";:FL=-1:RETURN
426 PRINT@845,"COLUMNA ";:FL=-2:RETURN
427 RETURN
428 Z3=256*(PEEK(16417))+PEEK(16416)-15360:RETURN
429 CLS:PRINT@25,"ENTRADA DE COLUMNA";:GOSUB379:CLS:GOTO430
430 GOSUB555:Y5=R1-1:Y6=R1-1:FORY1=C1TOC2:Y2=R2-R1:IFY2<10THENY2=1:GOTO431ELSEIFY2/10=INT(Y2/10)THENY2=Y2/10ELSEY2=INT(Y2/10)+1
431 FORY3=1TOY2:GOSUB528:Y7=64:Y9=128
432 Y4=Y4+1:IFY4>10THENY4=0:Y7=64:Y9=128:GOTO433ELSEY5=Y5+1:M(Y4)=Y5:IFY5>R2THENY8=0:Y4=0:Y7=64:Y9=128:GOTO433ELSEY8=Y5:Y7=Y7+64:GOS
UB539:GOSUB525:GOTO432
433 Y4=Y4+1:IFY4>10THENGOSUB435:GOTO434ELSEY6=Y6+1:IFY6>R2THENGOSUB435:GOTO434ELSEGOSUB436:GOSUB440:GOTO433
434 GOSUB437:IFIN$<>"N"THENGOSUB555:CLS:NEXTY3:Y6=R1-1:Y5=R1-1:CLS:NEXTY1:GOTO308ELSEP4=Y4-1:GOSUB554:X1=Y4:Y4=IN:X2=Y8:Y6=M(Y4):Y8=
M(IN):Y7=Y9+(64*IN)-64:GOSUB539:GOSUB552:GOSUB440:Y4=X1:Y8=X2:Y6=Y8:GOTO434
435 W2=(Y7-64)/64:W3=Y6-1:Y7=64:Y9=128:RETURN
436 Y8=Y6:Y7=Y7+64:GOSUB559:PRINT@Y7+45,"";:FL=9:GOSUB552:RETURN
437 PL=841:P5=40:GOSUB522:P4=10:GOSUB537:RETURN
438 T1=Y1:T8=Y8:Y8=T1:Y1=T8:RETURN
439 Y8=T8:Y1=T1:RETURN
440 IFIN$="/"THENA(Y1,Y6)=I:GOTO441ELSEIFIN$=""THEN442ELSEIFIN$="M"THEN308ELSEA(Y1,Y6)=IN
441 PRINT@Y7+26,"";:PRINTUSINGZ9$;A(Y1,Y8);:IFIN<>0THENI=IN
442 RETURN
443 CLS:PRINT@25,"ENTRADA DE ROTULOS":PRINT" ASEGURESE DE ROTULAR RENGLONES PARA S/TOTAL Y TOTAL":PRINT"LOS CALCULOS DEPENDERAN DE E
STOS ROTULOS":GOSUB379:CLS:GG$="ROTULOS DE RENGLONES":FORX=R1TOR2:X1=X1+1:GOSUB428:GOSUB447:PRINT@975,RW$(X);:SS=X
444 GOSUB450:IFIN$<>""THENRW$(SS)=IN$
445 PRINT@975,RW$(SS);:GOSUB451:NEXTX:PRINT@160,"ROTULADO COMPLETO";:X1=10:GOSUB447:IFC1=99ORC2=99THEN308ELSEQ3=3:CLS:GG$="ROTULADO
DE COLUMNAS":FORX=C1TOC2:X1=X1+1:GOSUB428:GOSUB447:PRINT@975,CM$(X);:SS=X:GOSUB450:IFIN$<>""THENCM$(SS)=IN$
446 PRINT@975,CM$(SS);:GOSUB451:NEXTX:PRINT@160,"END OF LABEL ENTRY";:X1=10:GOSUB447:CLS:Q3=0:GOTO308
447 IFX1<10THENRETURNELSEX1=0:PRINT@554,"EDITA UN ROTULO (S/N) ";:FL=1:GOSUB515:IFIN$<>"S"THENGOSUB452:RETURN
448 GOSUB452:PRINT@554,"          QUE ROTULO  ";:FL=-2:GOSUB515:IFQ3=3THEN449ELSEIFIN<R1ORIN>R2THEN448ELSEGOSUB452:SS=IN:GOSUB450:RW$(S
S)=IN$:PRINT@975,RW$(SS);:GOSUB451:X1=10:GOTO447
449 IFIN<C1ORIN>C2THEN448ELSEGOSUB452:SS=IN:GOSUB450:CM$(SS)=IN$:PRINT@975,CM$(SS);:GOSUB451:X1=10:GOTO447
450 PRINT@23,GG$;:PRINT@960,"";:PRINTUSING"##. ";SS;:FL=10:GOSUB515:RETURN
451 PRINT:PRINT@23,GG$;:RETURN
452 PRINT@554,"";:P5=22:GOSUB524:RETURN
453 CLS:PRINT@512,"INGRESE EL NOMBRE DE ESTE ARCHIVO ";:FL=8:GOSUB515:IFIN$=""ORVAL(IN$)>0THEN453ELSENA$=IN$:CLS:PRINT"INGRESE EL CO
MIENZO Y EL FINAL DE RENGLONES Y COLUMNAS QUE DESEA GRABAR";:GOSUB379:GOSUB454:NA$=NA$+DR$:OPEN"O",1,NA$:GOTO455
454 CLS:PRINT@512,"EN QUE DRIVE DESEA GRABAR LA PLANILLA PRINT ?";:FL=1:GOSUB515:IFIN$<"0"ORIN$>"3"THEN321ELSEDR$=":"+IN$:RETURN
455 PRINT#1,RW;CM;R1;R2;C1;C2:FORX=R1TOR2:PRINT#1,RW$(X):NEXT:FORX=C1TOC2:PRINT#1,CM$(X):NEXT:FORX=R1TOR2:X3=X3+1:CLS:PRINT@512,"GRA
BACION DE RENGLON ";:PRINTUSING"##";X;:FORX1=C1TOC2:PRINT#1,A(X1,X):NEXTX1,X:FORX=C1TOC2:PRINT#1,X:NEXT:CLOSE1:GOTO308
```

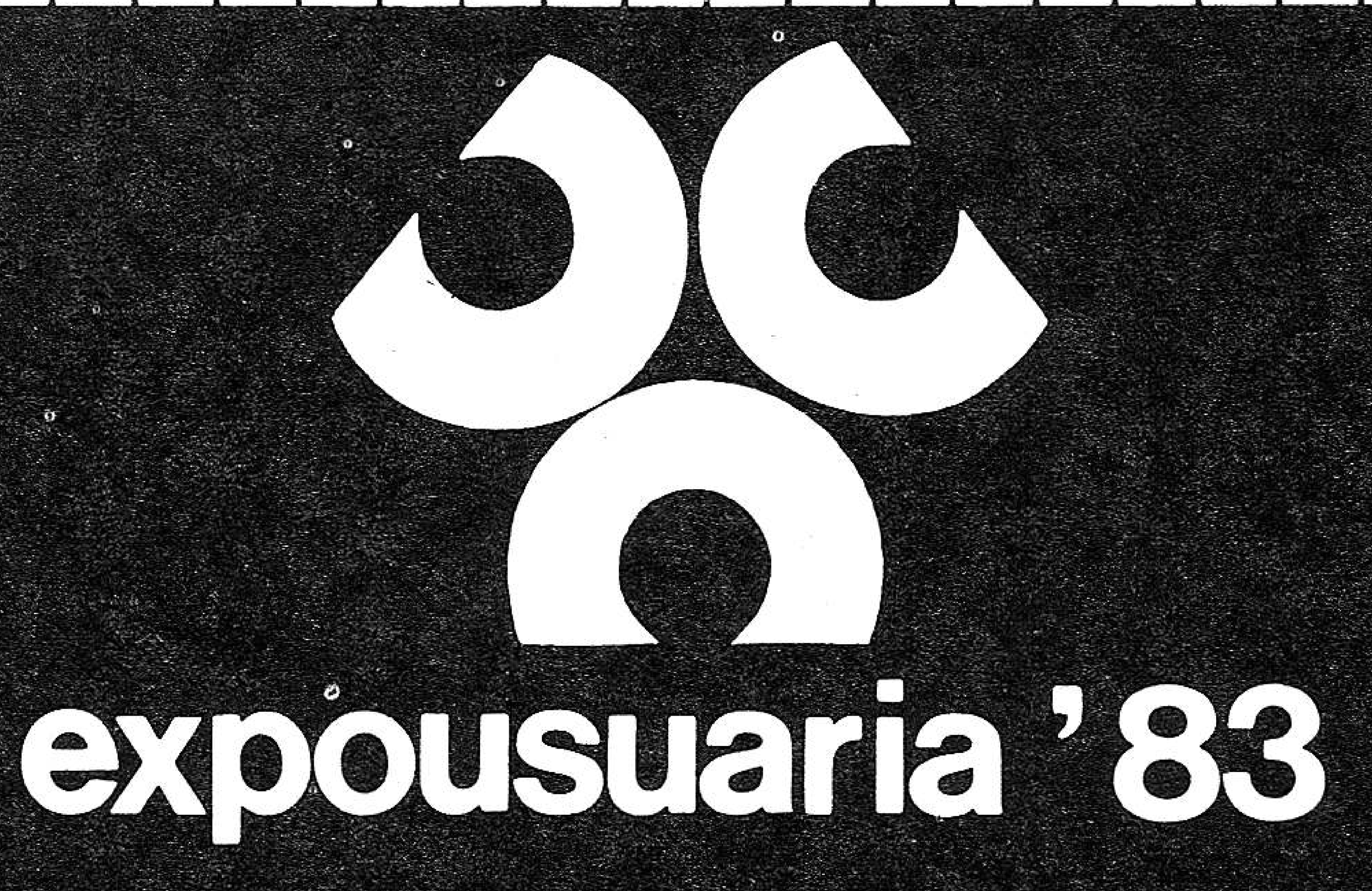
```
456 CLS:PRINT@512,"DESEA CONTINUAR (S/N) ";:FL=1:GOSUB515:IFIN$<>"S"THEN308ELSECLS:PRINT@512,"INGRESE EL NOMBRE DEL ARCHIVO ";:FL=8:
GOSUB515:KL$=IN$:CLS:PRINT@512,"ES ESTE UN ARCHIVO DE CALCULOS (S/N) ";:FL=1:GOSUB515
457 IFIN$="S"THENKL$=KL$+"/CALC"+DR$ELSEKL$=KL$+DR$
458 KILLKL$:GOTO308
459 CLS:PRINT@85,"LECTURA DE UNA PLANILLA":PRINT@256,"ESTA OPCION BORRA LOS DATOS DE LA MEMORIA ACTUALMENTE EN USO":PRINT@576,"DESEA
 CONTINUAR ? ";:FL=1:GOSUB515:IFIN$<>"S"THEN308
460 CLEAR0:CLEAR1000:ONERRORGOTO306:CLS:PRINT@512,"CUAL ES EL NOMBRE DE LA PLANILLA ? ";:FL=8:GOSUB515:IFIN$=""ORVAL(IN$)>0THEN308EL
SENA$=IN$:CLS:GOSUB484:CLS:GOSUB325:FORX=R1TOR2:INPUT#1,RW$(X):NEXT:FORX=C1TOC2:INPUT#1,CM$(X):NEXT:CLS
461 FORX=R1TOR2:PRINT@512,"LEYENDO EL RENGLON NRO.";X:FORX1=C1TOC2:INPUT#1,A(X1,X):POKE14305,15:NEXTX1,X:CLOSE:PRINT@512,"PLANILLA L
EIDA":FORX=1TO350:NEXT:GOTO308
462 CLS:PRINT@23,"LEER UN RENGLON":PRINT:PRINT"ESTA SUBRUTINA LE PERMITIRA LEER UN RENGLON DESDE OTRA PLANILLA":PRINT:PRINT"DESEA CO
NTINUAR (S/N) ";:FL=1:GOSUB515:IFIN$<>"Y"THENCLS:GOTO308ELSEGOSUB463:GOTO464
463 CLS:PRINT@512,"NOMBRE DE LA PLANILLA DONDE SOLICITA UN RENGLON ? ";:FL=8:GOSUB515:IFIN$=""ORVAL(IN$)>0THEN463ELSENA$=IN$:RETURN
464 PRINT@576,"QUE RENGLON NECESITA ";:FL=-2:GOSUB515:R3=IN:OPEN"I",1,NA$:INPUT#1,E1,E2,E3,E4,E5,E6:IFR3>E4THENCLOSE1:CLS:PRINT@576,
"ELECCION INVALIDA":FORX=1TO400:NEXT:CLS:GOTO464
465 IFCM<E2THENCLOSE:CLS:PRINT@512,"PLANILLA ";NA$;" ES MUY EXTENSA ":FORX=1TO400:NEXT:GOTO308
466 PRINT@640,"EN QUE RENGLON DESEARIA COLOCAR EL RENGLON ? ";:FL=-2:GOSUB515:IFIN>RWTHEN466ELSER4=IN:FORX=E3TOR3:INPUT#1,U$:NEXTX:R
W$(R4)=U$:FORX=R3+1TOE4:INPUT#1,U$:NEXT:FORX=E5TOE6:INPUT#1,U$:NEXT:CLS:PRINT@512,"BUSQUEDA DEL RENGLON NRO. ";
467 IFR3>E3THENFORX=E3TOR3-1:FORX1=E5TOE6:INPUT#1,U:POKE14305,15:NEXTX1:PRINT@526,X:NEXTX
468 CLS:PRINT@512,"NRO. DE COLUMNA EN QUE DESEA COMENZAR EL RENGLON ?":PRINT"PLANILLA ";NA$;" COMIENZA EN COL. NRO. ;E5;" HASTA ";E6
;" ";:FL=-2:GOSUB515:IFIN+E6>CMTHEN468ELSEX1=IN-E5:IFX1<0ORX1+E6>CMTHEN468ELSEFORX=E5TOE6:INPUT#1,A(X+X1,R4):NEXTX:CLS:CLOSE1
469 GOTO308
470 CLOSE:CLS:PRINT@23,"LEER UNA COLUMNA":PRINT:PRINT"ESTA RUTINA LE PERMITE LEER UNA COLUMNA DESDE OTRA PLANILLA":PRINT:PRINT"DESEA
 CONTINUAR (S/N) ";:FL=1:GOSUB515:IFIN$<>"S"THENCLS:GOTO308
471 CLOSE:CLS:PRINT@512,"DE QUE PLANILLA DESEA UN COLUMNA ? ";:FL=8:GOSUB515:IFIN$=" "ORVAL(IN$)>0THEN471
472 PRINT@576,"QUE COLUMNA DESEA ";:FL=-2:GOSUB515:R3=IN:OPEN"I",1,NA$:INPUT#1,E1,E2,E3,E4,E5,E6:IFR3>E6THENCLOSE1:CLS:PRINT@576,"EL
ECCION INVALIDA":FORX=1TO400:NEXT:CLS:GOTO472ELSEFORX3=E3TOE4:INPUT#1,U$:NEXT
473 IFRW<E1THENCLS:PRINT@512,"PLANILLA ";NA$;" ES MUY EXTENSA" :FORX=1TO400:NEXT:CLOSE1:GOTO308
474 PRINT@640,"EN QUE NRO. DE COLUMNA DESEA COLOCAR LA COLUMNA ? ";:FL=-2:GOSUB515:IFIN>CMTHEN474ELSER4=IN:FORX3=E5TOR3:INPUT#1,U$:N
EXTX3:CM$(IN)=U$:FORX3=R3+1TOE6:INPUT#1,U$:NEXTX3:FORX2=E3TOE4:CLS:PRINT@512,"LEYENDO EL RENGLON NRO. ";X2:WE=E5+R3-1
475 FORX3=E5TOWE:INPUT#1,U:NEXTX3:A(R4,X2)=U:POKE14305,15:FORX3=WE+1TOE6:INPUT#1,U:NEXTX3,X2:CLS:PRINT@512,"COLUMNA INGRESADA":FORX
=1TO400:NEXT:CLOSE1:GOTO308
476 Y5=0:Y6=0:CLS:PRINT@83,"INICIALIZACION DESDE OTRA PLANILLA":PRINT:PRINT:PRINT"DESEA CONTINUAR (S/N) ";:FL=1:GOSUB515:IFIN$<>"S"T
HEN476
477 CLEAR0:CLEAR1000:ONERRORGOTO306
478 CLS:PRINT@512,"NOMBRE DE LA PLANILLA DESDE DONDE DESEA INICIALAR ";:FL=8:GOSUB515:NA$=IN$:IFNA$=""ORVAL(NA$)>0THEN478ELSEGOSUB48
4:GOSUB325:CLS:PRINT@512,"DESEA ROTULO DE RENGLONES (S/N) ";:FL=1:GOSUB515:IFIN$="N"THENCLOSE:GOTO308ELSEPRINT
479 PRINT"DESEA ROTULOS DE RENGLONES  (S/N) ";:FL=1:GOSUB515:IFIN$="S"THENX5=1
480 PRINT:PRINT"DESEA ROTULOS DE COLUMNAS (S/N) ";:GOSUB515:IFIN$="S"THENX6=1
481 FORX=R1TOR2:IFX5=1THENINPUT#1,RW$(X)ELSEINPUT#1,U$
482 NEXTX:FORX=C1TOC2:IFX6=1THENINPUT#1,CM$(X)ELSEINPUT#1,U$
483 NEXTX:CLOSE:GOTO308
484 OPEN"I",1,NA$:INPUT#1,RW,CM,R1,R2,C1,C2:RETURN
485 QQ=0:FORX=1TORW:IFRW$(X)="TOTAL"THENIFQQ=0THENQQ=1:NEXTX
486 IFQQ>0ANDRW$(X)="TOTAL"THENCLS:PRINT@512,"MAS DE UN ROTULO DE TOTALES";:FORX=1TO500:NEXTX:CLS:GOTO308ELSENEXTX:CLS:Z=0:FORS=1TO9
:ONSGOSUB493,487,491,488,491,489,491,490,491:NEXTS:CLS:GOTO308
487 GG$="STOT":RETURN
488 GG$="STOT*":RETURN
489 GG$="STOT**":RETURN
490 GG$="TOTAL":RETURN
491 Y1=0:FORX5=1TORW-1:IFRW$(X5)=GG$THENY=Y1+1:Y1=X5-1ELSENEXTX5:RETURN
492 FORX1=1TOCM-1:A(X1,Y1+1)=0:FORY2=YTOY1:IFLEFT$(RW$(Y2),4)="STOT"THENNEXTY2,X1,X5:RETURNELSEA(X1,Y1+1)=A(X1,Y1+1)+A(X1,Y2):Z=Z+1:
PRINT@512,"CALCULO NRO. ";Z;:NEXTY2,X1,X5:RETURN
493 FORX=1TOCM:IFFU(X,13)=0THENNEXTX:RETURNELSEFORX1=1TORW:A(X,X1)=0:NEXTX1:FORX1=1TORW:J1=A(FU(X,13),X1):FORX2=1TO12STEP2:J3=FU(X,X
2):IFJ3=50RJ3=60RJ3=0THENX2=12:Z=Z+1:CLS:PRINT@512,"CALCULO NRO. ";Z;:A(X,X1)=J1:NEXTX2,X1,X:RETURN
494 J2=A(FU(X,X2+1),X1):J3=ABS(J3):IFJ3>40RJ3<1THENCLS:PRINT@512,"ERROR ENCONTRADO";:FORZZ=1TO500:NEXTZZ:CLS:GOTO308ELSEONJ3GOSUB495
,496,497,498:NEXTX2:Z=Z+1:CLS:PRINT@512,Z;:A(X,X1)=J1:NEXTX1,X:RETURN
495 J1=J1+J2:RETURN
496 J1=J1-J2:RETURN
497 J1=J1*J2:RETURN
498 IFJ1=0ANDJ2=0THENRETURNELSEIFJ1=0ORJ2=0THENCLS:PRINT@512,"ERROR - DIVISION POR 0 EN COLUMNA NRO. ";X:FORZ0=1TO500:NEXTZ0:PRINT@9
91," ";:FL=2:GOSUB515:GOTO308ELSEJ1=J1/J2:RETURN
499 CLS:PRINT@15,"PANTALLA Y EDICION";:PRINT@512,"COMENZANDO RENGLON ";:FL=-2:P4=RW:GOSUB548:Y3=IN:PRINT@537,"  COMENZANDO COLUMNA "
;:P4=CM:GOSUB548:Y2=IN:CLS:GOTO504
500 Y1=1:Y2=1
501 Y1=PEEK(14400):IFY1<>0THEN502ELSE501
502 Y9=PEEK(14464):IFY9=1THEN503ELSEIFY1=8THENY3=Y3+1:GOTO504ELSEIFY1=16THENY3=Y3-1:GOTO504ELSEIFY1=32THENY2=Y2+1:GOTO504ELSEIFY1=64
THENY2=Y2-1:GOTO504
```

```
503 IFY1=8ANDY9=1THENY3=Y3+9:GOTO504ELSEIFY1=16ANDY9=1THENY3=Y3-9:GOTO504ELSEIFY1=32ANDY9=1THENY2=Y2+3:GOTO504ELSEIFY1=64ANDY9=1THEN
Y2=Y2-3:GOTO504ELSEIFY1=128THEND=2:GOTO511ELSE501
504 CLS:IFY2<1THENY2=1
505 IFY2>CM-4THENY2=CM-4:GOTO504
506 IFY3<1THENY3=1
507 IFY3>RW-10THENY3=RW-10:GOTO506ELSEY4=Y2+3:IFY4>CMTHENY4=CM
508 Y5=Y3+9:IFY5>RWTHENY5=RW
509 M(1)=138:M(2)=151:M(3)=164:M(4)=177:Y7=128:FORX1=Y2TOY4:X3=X1-Y2+1:PRINT@M(X3)-125,"COLUMNA NRO.";X1;:PRINT@M(X3)-61,CM$(X1);:NE
XTX1:FORX1=Y3TOY5:X2=X1-Y3:PRINT@Y7+((X2+1)*64),"";:PRINTUSING"##. ";X1;:PRINTRW$(X1);:NEXTX1:FORX=Y3TOY5
510 FORX1=Y2TOY4:X2=X1-Y2+1:X3=X-Y3+1:PRINT@M(X2)+(X3*64),"";:PRINTUSINGZ9$;A(X1,X);:IFPEEK(14400)<>0THENY1=PEEK(14400):GOTO502ELSEN
EXTX1,X:IFD=2THEN511ELSEPRINT@960,"USE FLECHAS Y <SHIFT>FLECHAS PARA DESPLAZAMIENTOS - BARRA PARA OPCIONES";:GOTO501
511 D=0:PL=960:P5=63:GOSUB522:PRINT@960,"INGRESE 1 (IMPRIME PANTALLA)- 2 (RETORNA A MENU)- 3 (EDITA) ";:FL=-1:P4=3:GOSUB548::IFIN=1T
HENGOSUB518:GOTO501ELSEIFIN=2THENCLS:GOTO308ELSEGOSUB522:PRINT@960,"EDITAR > 1-COLUMNA ROTULADA 2-RENGLON ROTULADO O 3-NRO. "
512 FL=-1:GOSUB548:IFIN=1THENGOSUB522:PRINT@960,"INGRESE EL NRO. DE COLUMNA ";:FL=-2:P4=CM:GOSUB548:X=IN:PRINT@985,"  INGRESE EL NUE
VO ROTULO ";:FL=10:GOSUB515:CM$(X)=IN$:GOTO504
513 IFIN=2THENGOSUB522:PRINT@960,"INGRESE EL NRO.DE RENGLON";:FL=-2:P4=RW:GOSUB548:X=IN:PRINT@985,"INGRESE NVO.ROTULO";:FL=10:GOSUB5
15:RW$(X)=IN$:GOTO504ELSEGOSUB522:PRINT@960,"INGRESE NRO.DE RENGLON ";:FL=-2:P4=RW:GOSUB548:R=IN:PRINT@985,"COLUMNA NRO.";
514 P4=CM:GOSUB548:C=IN:PRINT" NUEVO NUMERO  ";:FL=9:GOSUB552:A(C,R)=IN:GOTO504
515 VV=INT(ABS(FL)):V1=0:IFABS(FL)-VV<>0THENV1=1
516 V1=1DEFUSR0=&HFE00:Y=USR0(VV):DEFUSR1=&HFE08:Y=USR1(V1):V1=1:IFFL<0THENV1=0
517 IN=0DEFUSR2=&HFE10:Y=USR2(V1):IN$=STRING$(VV," ")::DEFUSR3=&HFE18:Y=USR3(VARPTR(IN$)):IFFL>0THENRETURNELSEIN=VAL(IN$):RETURN
518 RETURN:DEFUSR0=&HFD7A:VV=USR0(0):RETURN
519 PRINT@PL,"INGRESE EL NRO. DE LINEA A CORREGIR ";:FL=-2:RETURN
520 FORX=1TOX1:NEXT:RETURN
521 PRINT@PL,"ELECCION INVALIDA";:RETURN
522 PRINT@PL,STRING$(P5," ");:RETURN
523 PRINTSTRING$(P5,CHR$(CH));:RETURN
524 PRINTSTRING$(P5," ");:RETURN
525 JL=ABS(FL):PRINTSTRING$(JL,".");:RETURN
526 IFRW$(Y1)=""THENCLS:PRINT@512,"RENGLON NRO. ";Y1;"NO TIENE ROTULO INGRESELO AHORA ";:FL=10:GOSUB515:RW$(Y1)=IN$:CLS:GOTO526ELSEP
RINT@10,"INGRESE LAS COLUMNAS PARA EL RENGLON NRO. ";Y1;" ROTULADO ";RW$(Y1);:PRINT@68,"COL NRO.":PRINT@75,"ROTULO",
527 PRINT@96,"CONTENIDO";:PRINT@109,"INGRESO";:RETURN
528 GOSUB529:PRINT@10,"INGRESE LOS RENGLONES PARA LA COLUMNA NRO. ";Y1;" ROTULADA ";CM$(Y1);:PRINT@68,"RENG.NRO.;:PRINT@75,"ROTULO";
:PRINT@96,"CONTENIDO",:PRINT@109,"INGRESO";:RETURN
529 IFCM$(Y1)=""THENCLS:PRINT@512,"COL.NRO.";Y1;"ESTA SIN ROTULAR. DESEA ROTULARLA (S/N) ";:FL=1:GOSUB515:IFIN$="N"THENCLS:RETURNELS
EPRINT:PRINT"INGRESE EL ROTULO ";:FL=10:GOSUB515:CM$(Y1)=IN$:CLSELSERETURN
530 PRINT@394,"2. INGRESE EL NUMERO DE RENGLONES ";:FL=-2:RETURN
531 PRINT@458,"3. INGRESE EL NUMERO DE COLUMNAS ";:FL=-2:RETURN
532 PRINT@330,"1. INGRESE EL NOMBRE DE ESTA PLANILLA ";:FL=8:RETURN
533 PRINT@280,"1. COMIENZO EN COLUMNA ";:FL=-2:RETURN
534 PRINT@344,"2.FINALIZA EN COLUMNA ";:FL=-2:RETURN
535 PRINT@408,"3. COMIENZO EN RENGLON ";:FL=-2:RETURN
536 PRINT@472,"4. FINALIZA EN RENGLON";:FL=-2:RETURN
537 PRINT@PL,"LOS DATOS SON CORRECTOS (S/N) ";:FL=1:M3$=" ":M1$="S":M2$="N":GOSUB549:IFF=2THEN537ELSEIFIN$="N"THENF=2:RETURNELSERETU
RN
538 PRINT@Y7,"";:PRINTUSING"##.  ";Y4;Y8;:PRINTCM$(Y8);:PRINT@Y7+26,"";:PRINTUSINGZ9$;A(Y8,Y1);:PRINT@Y7+45,"";:FL=9:RETURN
539 PRINT@Y7,"";:PRINTUSING"##.  ";Y4;Y8;:PRINTRW$(Y8);:PRINT@Y7+26,"";:PRINTUSINGZ9$;A(Y1,Y8);:PRINT@Y7+45,"";:FL=9:RETURN
540 PRINT@512,"SUMAR UN ";GG$;" DESPUES DE ";GG$;" ";:FL=-2:RETURN
541 PRINT@512,"1. DESPLAZAR ";GG$;" ";:FL=-2:RETURN
542 PRINT@542,"2. DESPLAZARLO A ";GG$;" ";:FL=-2:RETURN
543 PRINT@512,"COLUMNA CONSTANTE ";:FL=-2:RETURN
544 PRINT@546,"INGRESE EL MONTO ";:FL=-8:RETURN
545 PRINT@576,"DESDE RENGLON NRO. (1-";RW;")";:FL=-2:RETURN
546 PRINT@603,"AL RENGLON NRO. (";Y4;" - ";RW;")";:FL=-2:RETURN
547 PRINT@512,"CUAL ";GG$;" DESEA BORRAR (1 -";Y2;") ";:FL=-2:RETURN
548 GOSUB428:GOSUB515:IFIN<1ORIN>P4THENPRINT@Z3,"";:GOTO548ELSERETURN
549 GOSUB515:IFIN$="M"THEN308ELSEIFM1$=IN$THEN550ELSEIFM2$=IN$THEN550ELSEIFM3$=IN$THEN550ELSEF=2:RETURN
550 F=1:RETURN
551 M3$=" ":M1$="S":M2$="N":RETURN
552 GOSUB428:GOSUB515:IN=VAL(IN$):IFIN<VL*-1ORIN>VLTHEN553ELSEPRINT@Z3,STRING$(9," ");:RETURN
553 PRINT@Z3,"";:GOTO552
554 PRINT@PL,"";:P5=40:GOSUB522:GOSUB519:GOSUB548:RETURN
555 Y4=0:Y9=0:RETURN
556 CLS:PRINT@512,"INGRESE EL NRO. DE COLUMNA PARA BORRAR LOS CALCULOS ";:FL=-2:P4=CM:GOSUB548:FORX=1TO13:FU(IN,X)=0:NEXT:GOTO308
557 IFPEEK(14312)=63ORPEEK(16421)<>6THENRETURNELSEPRINT@960,"LA IMPRESORA NO ESTA LISTA - <ENTER> PARA CONTINUAR ";:FL=1:GOSUB515:IF
IN$="M"THEN308ELSE557
558 IFCM$(Y8)=""THENFL=10:PRINT@Y7+10,"";:GOSUB515:CM$(Y8)=IN$:RETURNELSERETURN
559 IFRW$(Y8)=""THENFL=10:PRINT@Y7+10,"";:GOSUB515:RW$(Y8)=IN$:RETURNELSERETURN
560 CLS:PRINT@512,"INGRESE LA COLUMNA FUENTE ";:FL=-2:GOSUB515:IFIN<1ORIN>CMTHEN560ELSESU=IN:PRINT:PRINT"INGRESE LA COLUMNA DESTINO
";:FL=-2:GOSUB515:IFIN<1ORIN>CMTHEN560ELSEDS=IN
561 PRINT@832,"ESTE PROGRAMA PROVEERA UN TOTAL ACUMULATIVO DE LA COLUMNA FUENTE EN LA COLUMNA DESTINO.":PRINT"DIGITE <ENTER> PARA LA
 PERFORMANCE O <M> PARA MENU ";:FL=1:GOSUB515:IFIN$="M"THEN308ELSEFORX=1TORW
562 IFRIGHT$(RW$(X),4)="STOT"ORA(SU,X)=0THENNEXTXELSEIFRW$(X)="TOTAL"THEN308ELSEA8=A(SU,X)+A8:A(DS,X)=A8:NEXTX
563 A8=0:GOTO308
564 DW=PEEK(16425):DW=PG-DW+1:FORDQ=1TODW:LPRINT" ":NEXTDQ:POKE16425,1:RETURN
565 BOSUB344:GOTO316
566 GOSUB515:IFIN<1ORIN>100THEN566ELSEIN%=IN:POKE(&HFC25),IN%:RETURN
```

LA SEMANA MAS IMPORTANTE EN EL MUNDO DE LA INFORMATICA

19 al 23 de abril
Sheraton Hotel

1ª exposición de equipamientos técnicas y servicios para la informática

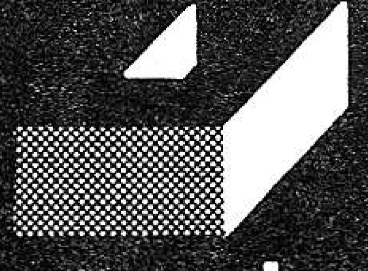


expousuaria '83

Una muestra integral de todo cuanto interesa a usuarios, técnicos, científicos e industriales vinculados a la informática a través de las siguientes empresas participantes

HASLER ARGENTINA S.A.	stand 1-2
LANGENAUER Y CIA.	3-4
ARGECINT S.R.L.	5-6
AUTOM S.R.L.	7
PLUS COMPUTER S.A.	8-9-10-55
CIA. BURROUGHS DE MAQUINAS LTDA.	11-12-13
INTER SEAS S.A.	14-15
INPUT S.R.L.	16
SERVOTRON S.A.C.I.F.I.	17
CONSAD S.A.	18
CFB S.A.	19
TEXAS INSTRUMENTS S.A.C.I.F.	20-21-22
COSTOS Y ORGANIZACION S.A.	23
IDSA S.R.L.	24
HEWLETT PACKARD ARGENTINA S.A.	25
SISTEMAS ELECTRONICOS DE SEGURIDAD S.A.	26
PAN AMERICAN COMPUTER SYSTEMS	27
SISTECO S.A.	28-29-30
PROCEDA S.A.	31-32-33
SYSCOM S.R.L.	34
BASF ARGENTINA S.A.	35
PHILIPS ARGENTINA S.A.	36
SERVICIOS EN INFORMATICA S.A.	37
DATA MEMORY S.A.	38
FACEMA S.A.	39
BULL ARGENTINA S.A.	40-41-42
COASIN COMPUTACION S.A.	43-44
IBM ARGENTIÑA S.A.	45-46-47
APD S.A.	48
CRAFTING SERVICE S.A.	49
C. W. COMUNICACIONES - COMPUTER WOLD	50-51
MC. CORMACK DODGE	52
MINI COMPUTER	53
ROTATIVOS VENUS S.A.	54

En forma simultánea, 1er. Congreso Nacional de Informática y Teleinformática, y otras Jornadas, Encuentros y Seminarios sobre el tema.



Promueve **usuaria**
Asociación Argentina de Usuarios de la Informática



organiza
Inforexco SRL.
Miembro de AEFA
H. Yrigoyen 1427 - 9º piso
1089 Bs. As. - Tel. 37-5399/9964

La Exposición estará abierta al público de 10 a 21 hs.
Horarios estudiantes, de 10 a 11,30 y de 14 a 15.30

MARQUE CON UN CIRCULO EL 77 EN LA SERVICE CARD

UN TERMINAL DEL RS-232C MODELO I/III

Segunda Parte

Ing. Alfredo López Quinteros

Antes de proporcionarle ejemplos reales de programación sobre operaciones en serie, vamos a describir como se construye una plaqueta de conexiones RS-232C. Esta plaqueta de conexiones se puede utilizar en experimentos con los ejemplos de programación que le vamos a indicar. Algo muy importante, puede utilizarse para ayudarle a interconectar dispositivos en serie para su computadora Modelo I ó III. La plaqueta de conexiones se indica en la Figura 8.

La plaqueta interrumpe un cable RS-232C de 25 líneas y rutea las líneas a través de una plaqueta prototipo. Se utilizan la plaqueta de longitudes cortas de cable sólido calibre 20 para conectar ambos lados del cable, tal como se ilustra en la figura 9. Para transponer RD y TD, a modo de ejemplo, simplemente entrecruce las patas 2 y 3 tal como se indica en la figura.

En forma similar se pueden conectar otras líneas. Para probar el estado de cualquier línea, se puede co-

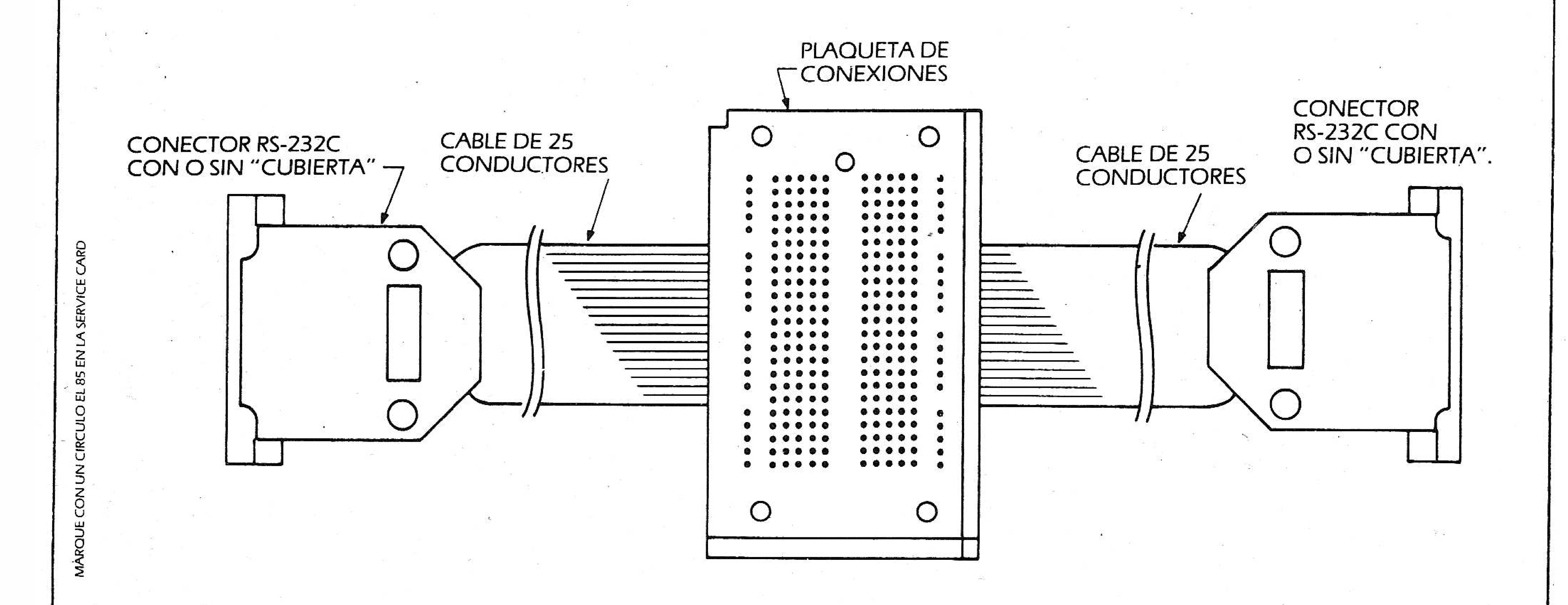


Figura 8: La plaqueta de conexiones para comunicaciones de información utiliza una plaqueta prototipo para "interrumpir" las 25 líneas de un cable RS-232C. Esto permite que las líneas puedan ser intercambiadas, puenteadas o probadas.

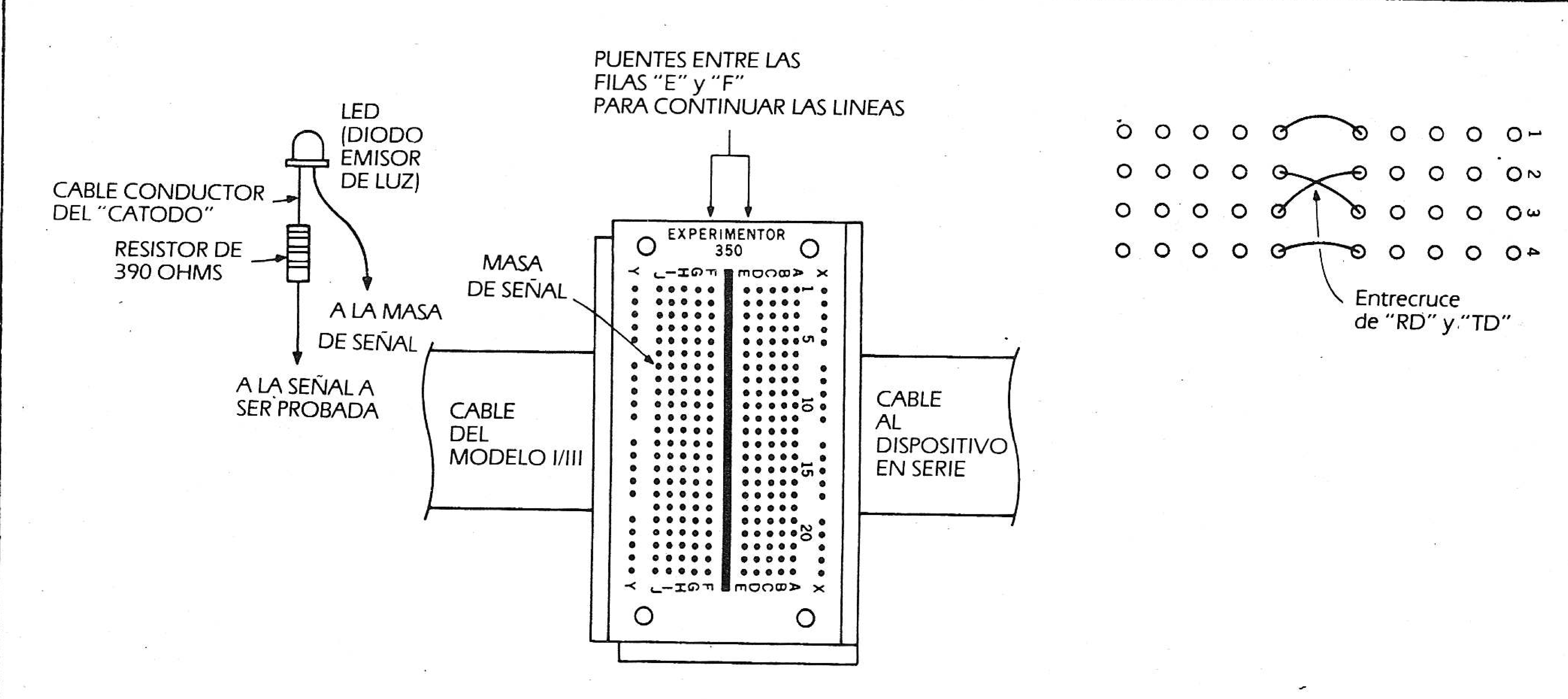


Figura 9: Para probar todas las líneas, puentear cada fila desde los puntos "E" a "F". Las líneas pueden ser fácilmente interconectadas. Se puede utilizar un LED con un resistor en serie para probar el estado de cualquiera de las líneas tanto sea en el lado de entrada como en el de salida.

nectar un LED con un resistor a la señal de masa, pata 7, y la línea en cuestión, tal como se ilustra en la figura. Las líneas se pueden "simular" en cambio de realizar el cableado a una conexión especial RS-232C agregando una modificación entre una línea activa y la línea a simularse.

Para construir la plaqueta, utilice la versión más pequeña de la plaqueta prototipo Radio Shack (catálogo 276-175). La parte posterior tiene una cubierta de papel engomado. Quítela para dejar al descubierto los flejes de interconexión, tal como se ilustra en la figura 10. Las hileras están numeradas del 1 al 23 en la parte frontal del tablero.

Usted necesitará dos conectores RS-232C. Obtenga el tipo para soldar, no así el tipo de desplazamiento de aislación. Desafortunadamente Radio Shack vende la del último tipo. Utilice las combinaciones de conexiones adecuadas para el equipo que está interconectando. Los modems, por ejemplo, generalmente tienen conectores RS-232C hembra, y requieren un conector macho en el cable.

Aunque las patas del conector están numeradas, dicha numeración frecuentemente es dificultosa en lo que a su lectura se refiere a menos que se sostenga el conector en el ángulo correcto y con la luz adecuada. La numeración standard para los conectores RS-232C que aparece en el mismo, se ilustra en la figura 11.

Obtenga un cable flexible para 25 alambres (Radio Shack, catálogo 278-771: es un cable de 40 conductores que se puede separar). Pase los alambres por el cable de manera tal que las posiciones de los alambres correspondan a los números de las patas RS-232C.

QUITE LA PROTECCION ADHESIVA

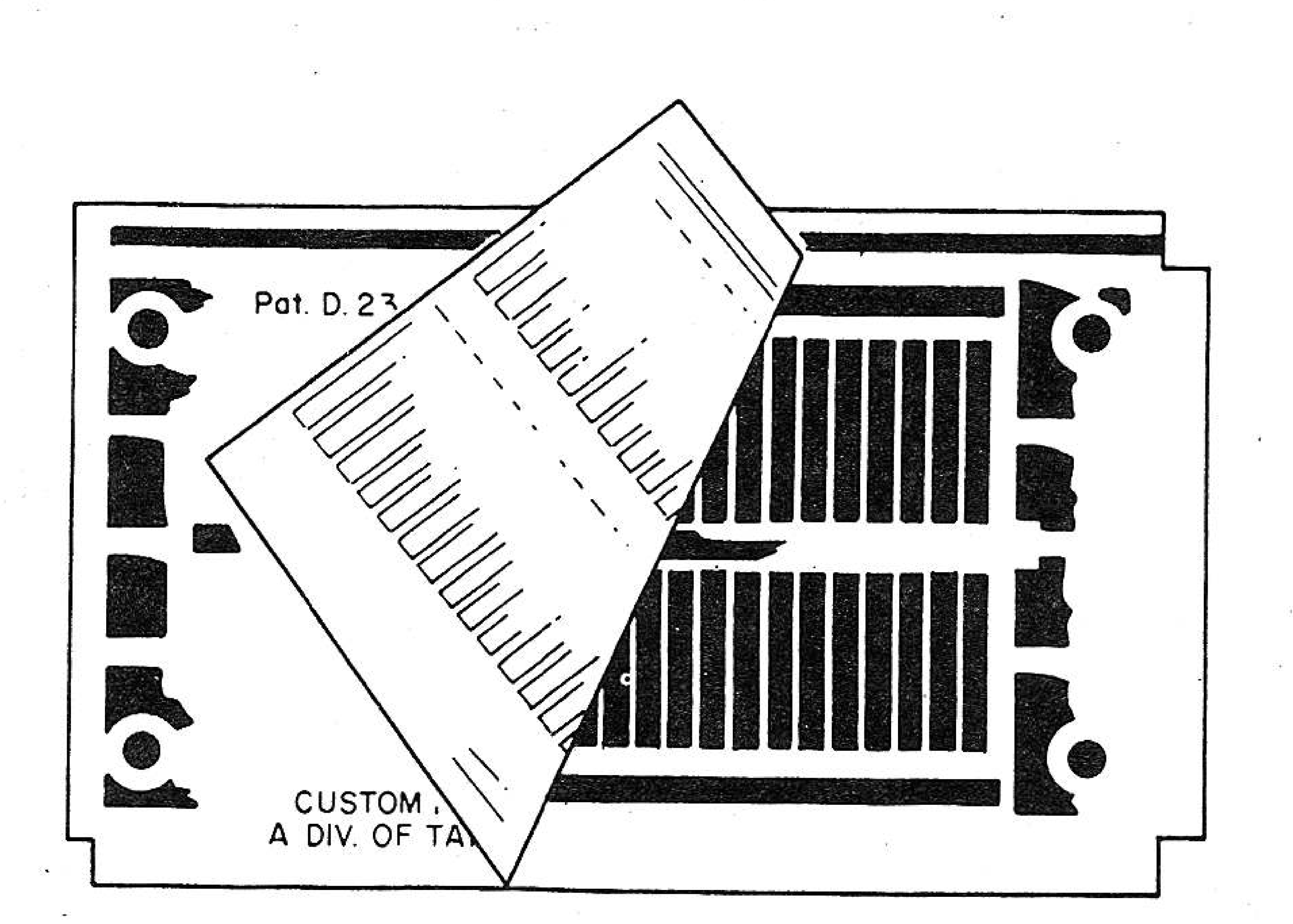


FIGURA 10: Quite la protección de la plaqueta prototipo para dejar al descubierto las láminas para soldar.

1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21 22 23 24 25
CONECTOR RS-232C MACHO -DB-25P- VISTA FRONTAL

13 12 11 10 9 8 7 6 5 4 3 2 1
25 24 23 22 21 20 19 18 17 16 15 14
CONECTOR RS-232C HEMBRA -DB-25S- VISTA FRONTAL

FIGURA 11: Numeración de las patas del conector RS-232C standard.

Esto significará separar el cable en dos mitades, tal como se ilustra en la figura 12. Corte el cable RS-232C por la mitad. Separe los alambres en ambos extremos del cable y separe cada alambre aproximadamente 1/16" con un separacable.

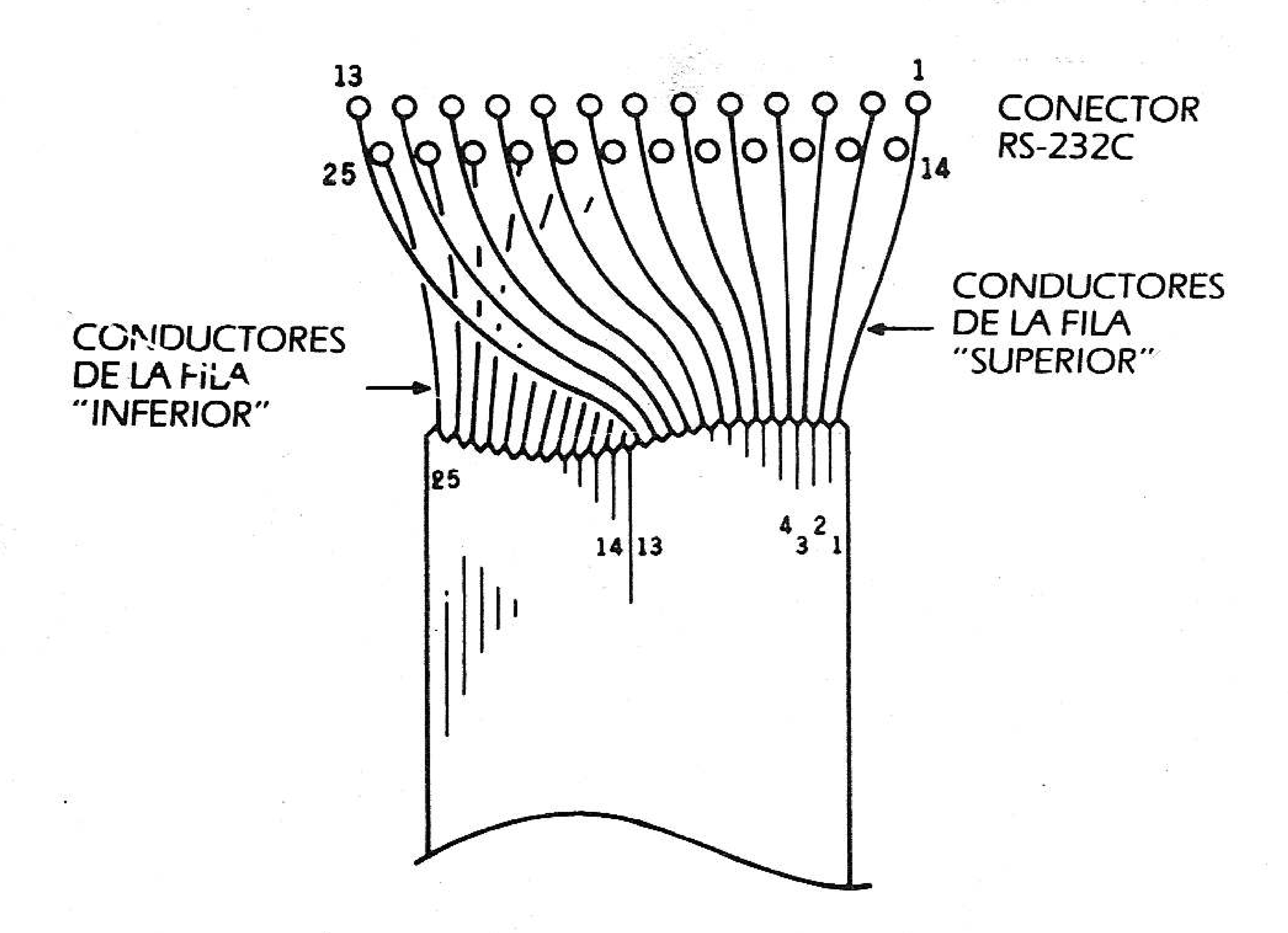


FIGURA 12: Pasando los alambre al cable. El cableado no es standard comparado a la mayoría de los cables RS-232C y correlaciona los números de las patas de RS-232C con la numeración de posición del conductor.

Realice la soldadura de los 23 alambres en las 23 hileras del tablero. Haga coincidir el número de fila con los números de patas RS-232C. Proceda a la soldadura del alambre 24 (pata 24) al fleje vertical en cada lado de la plaqueta. Cortar el cable 25. Esta señal no está definida en la especificación RS-232C (ver figura 13).

Utilizando un alambre calibre 20, conectar las patas opuestas para aquellas líneas que no van a cambiarse. El cable se adecuará correctamente en los agujeros de la plaqueta y permanecerá allí. Sólo las señales del Modelo I/III indicadas en la Tabla 1 necesitan ser conectadas de esta manera. No se utilizan las restantes.

Ahora usted ya está listo para experimentar con los ejemplos de programación indicados seguidamente o utilizar la plaqueta de conexión para ayudar a conectar los dispositivos en serie a su sistema.

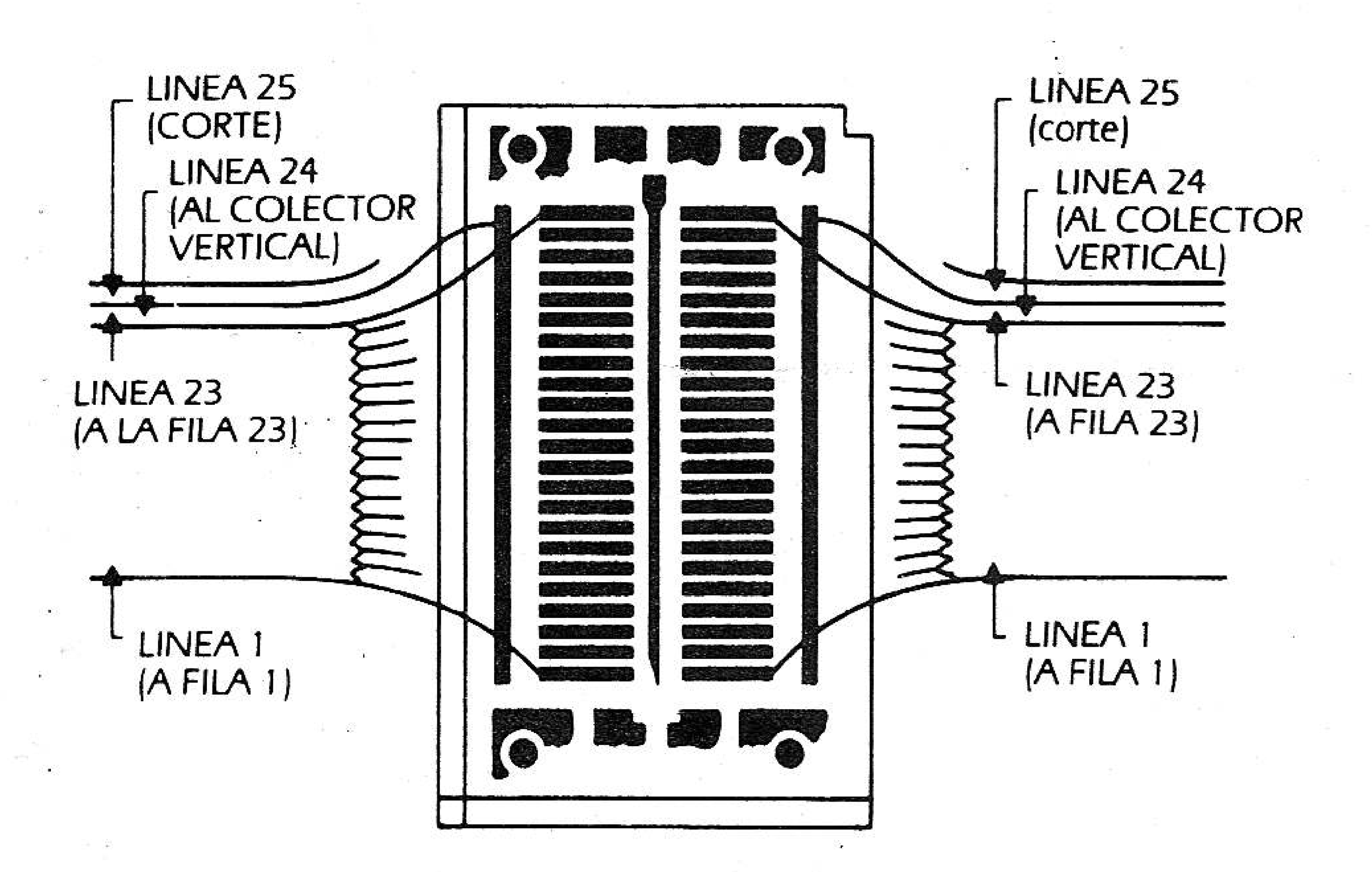


FIGURA 13: Soldar el cable a la plaqueta prototipo. Se sueldan 23 líneas a las filas 1-23. La línea 24 se suelda a la columna vertical. La línea 25, no utilizada, se puede cortar.

```
90 'EJERCITACION DE SALIDA DE DATOS
100 OUT 232,0
105 PRINT "INGRESE EP, WLS, SSB, PI"
110 INPUT EP,WL,SS,PI
120 WD=EP*128+WL*32+SS*16+PI*8+4
130 OUT 234,WD
135 INPUT "CODIGO DE BAUDIO";BA
150 OUT 233,BA
160 INPUT "CODIGO DE CARACTER";CH
165 I=0
170 A=INP(234)
180 IF (A AND 64)=0 THEN GOTO 170
185 I=I+1
190 OUT 235,CH
200 GOTO 170
```

LISTADO 1: Este simple programa BASIC continuará emitiendo datos sobre la línea TD a las velocidades de bits y formatos definidos por el usuario.

Ejemplos de programación de RS-232C

Los ejemplos indicados a continuación están en BASIC. Pueden convertirse a lenguaje assembly Z80 substituyendo las ENTRADAS y SALIDAS para las ENTRADAS y SALIDAS de BASIC. No hay espacio para indicar las impresoras en serie o los accionamientos de las comunicaciones, pero estos ejemplos develarán el misterio de la interconexión RS-232C.

Fijación de RTS, DTR, e Interrupción

Las líneas RTS y DTR son salidas del controlador RS-232C que indican "pedido para enviar" y "terminal de datos de lista". La línea RTS (pata 4) se fija de la siguiente manera:

```
100   OUT 232,0          inicializa RS-232C
110   OUT 234, xxxxxxxl  valor binario con fijación lsb
```

("xxxxxxx1" indica que se debe fijar el bit 0; los restantes bits deberán fijarse o reponerse adecuadamente).

La línea DTR (pata 20) se fija en forma similar:

```
100   OUT 232,0          inicializa RS-232C
110   OUT 234,xxxxxxlx   valor binario
```

Si se deben fijar ambos RTS y DTR, el valor binario sería xxxxxx11.

La "interrupcón" no proviene del conector RS-232C pero habilita o inhabilita la línea TD. Utilice un 1 para habilitar la línea TD:

```
100   OUT 232,0          inicializa RS-232C
110   OUT 234,xxxxxlxx   habilita TD
```

Utilice el código anteriormente indicado y experimente con la plaqueta conectando un LED y un resistor de 390 ohm entre la pata 7 de la plaqueta y la pata 4 ó

20. Recuerde que la lógica 1 es −12 VCC y la lógica 0, +12 VCC.

Lectura de CTS, DSR, CD y RI

Estas entradas al controlador RS-232C indican libre para enviar, fijación de información lista, detección de portación e indicador de anillo. Para leer las líneas realice:

```
100   OUT 232,0        inicializa RS-232C
110   A = INP(232)     lee las líneas
```

La variable A contendrá un valor binario correspondiente a CTS, DSR, CD y RI en los 4 bits más importantes, los 4 bits inferiores no son importantes.

Usted puede experimentar fijando primero o RTS o DTR y puenteando en la plaqueta entre las patas RTS o DTR a las cuatro líneas de entrada. El puente es la forma más sencilla de "simular" una señal, tanto mediante el cableado del conector o, en este caso, en la plaqueta.

```
90 'EJERCITACION DE RETROVERIFICACION DE LAZO
100 OUT 232,0
105 PRINT "INGRESE EP, WLS, SSB, PI"
110 INPUT EP,WL,SS,PI
120 WD=EP*128+WL*32+SS*16+PI*8+4
130 OUT 234,WD
135 INPUT "CODIGO DE BAUDIOS";BA
150 OUT 233,BA
160 INPUT "CODIGO DE CARACTER";CH
165 I=0
170 A=INP(234)
180 IF (A AND 64)=0 THEN GOTO 170
185 I=I+1
190 OUT 235,CH
200 A=INP(234)
210 IF (A AND 128)=0 THEN GOTO 170
220 A=INP(235)
230 PRINT A
240 GOTO 170
```

LISTADO 2: Programa de ejercitación de retroverificación de lazo o circuito en BASIC emite la información que es ingresada inmediatamente a través de la línea RD. Las patas 2 y 3 de las líneas RS-232C deberán ser punteadas en forma conjunta.

Fijación de SUN, STD y SRTS

Estas líneas secundarias normalmente no se utilizan en los programas de comunicaciones, pero se las puede fijar de la siguiente manera:

```
100   OUT 232,0            inicializa RS-232 C
110   A = INP (233)        flip-flop (circuito biestable)
                           CRL basculante
120   OUT 234,xxUTRxxx     fija SUN, STD, SRTS
130   A = INP(233)         CRL basculante
```

En la línea 120, los bits 5, 4 y 3 controlan el estado de SUN, STD y SRTS, respectivamente. INP (233) mueve (acciona) el circuito biestable de carga del registrador de control de manera tal que el registrador de control no se cargue.

Salida en TD

El código BASIC en el listado 1 indica una salida continua de un caracter específico. Si usted posee un osciloscopio, conéctelo entre la pata 7 (SGND) y la pata 2 (TD) y observar la salida.

ACA VA LISTADO 1

ACA VA LISTADO 2

Las entradas EP, WL, SS y PI definen paridad par/impar, longitud de palabra, número de bits de detención, e inhibición de paridad, respectivamente.

Esta información con formato se envía al registrador de control mediante "OUT 234, WD".

El código de velocidad de bit (ver tabla 2) se envía a BRG mediante "OUT 233, BA".

El valor del carácter (CH) es el código decimal para el carácter que se va a enviar. Este valor debe estar en un rango de 0-255. ASCII "A", por ejemplo, es decimal 65.

El lazo en 170-200 verifica continuamente el estado THRE. Si el transmisor que mantiene el registrador esta vacío, emite como salida el carácter al transmisor que mantiene el registro mediante un "OUT 235, CH".

La figura 14 ilustra las formas de onda de un osciloscopio para CH = 65 y CH = 85 sin bit de paridad, 1 bit de detención, 8 bits de información y 300 bps.

Este lazo BASIC opera correctamente con una velocidad de 30 caracteres por segundo. Fijando la velocidad de bits en 600 (BA = 102) dio como resultado un rendimiento de 41 cps (no 60 a consecuencia de las desventajas de BASIC). Es posible accionar una impresora en línea con BASIC.

La última aplicación se ilustra en el listado 2. Este es un ensayo de retroverificación en el cual la salida de la línea TD se puentea nuevamente a la entrada de línea RD. El carácter enviado retorna en la línea RD. Esta técnica es común para probar una interconexión RS-232C localmente. Puentea las dos líneas mediante un alambre corto entre las patas 2 y 3 en la plaqueta.

Conexión de dispositivos en serie

Se dispone de tantos dispositivos en serie que resulta difícil generalizar sobre cuales son las configuraciones adecuadas del cable RS-232C para accionar estos dispositivos. Una típica configuracion que se puede fijar por intermedio de la plaqueta se ilustra en la figura 15. Esto conecta la Spinwriter NEC al Modelo I ó III y permite velocidades de transferencia de hasta 1.200 bps.

Lea los requerimientos de interconexión para el dispositivo en serie que se va a utilizar con su Modelo I ó III, utilice la plaqueta para probar las condiciones de línea , y posiblemente aún utilizar alguno de los códigos suministrados anteriormente. Este método facilitará lainterconexión con equipos RS-232C que no sean Radio Shack y le ayudará a darse cuenta de los beneficios implícitos en el slogan publicitario "compatible con RS-232C".

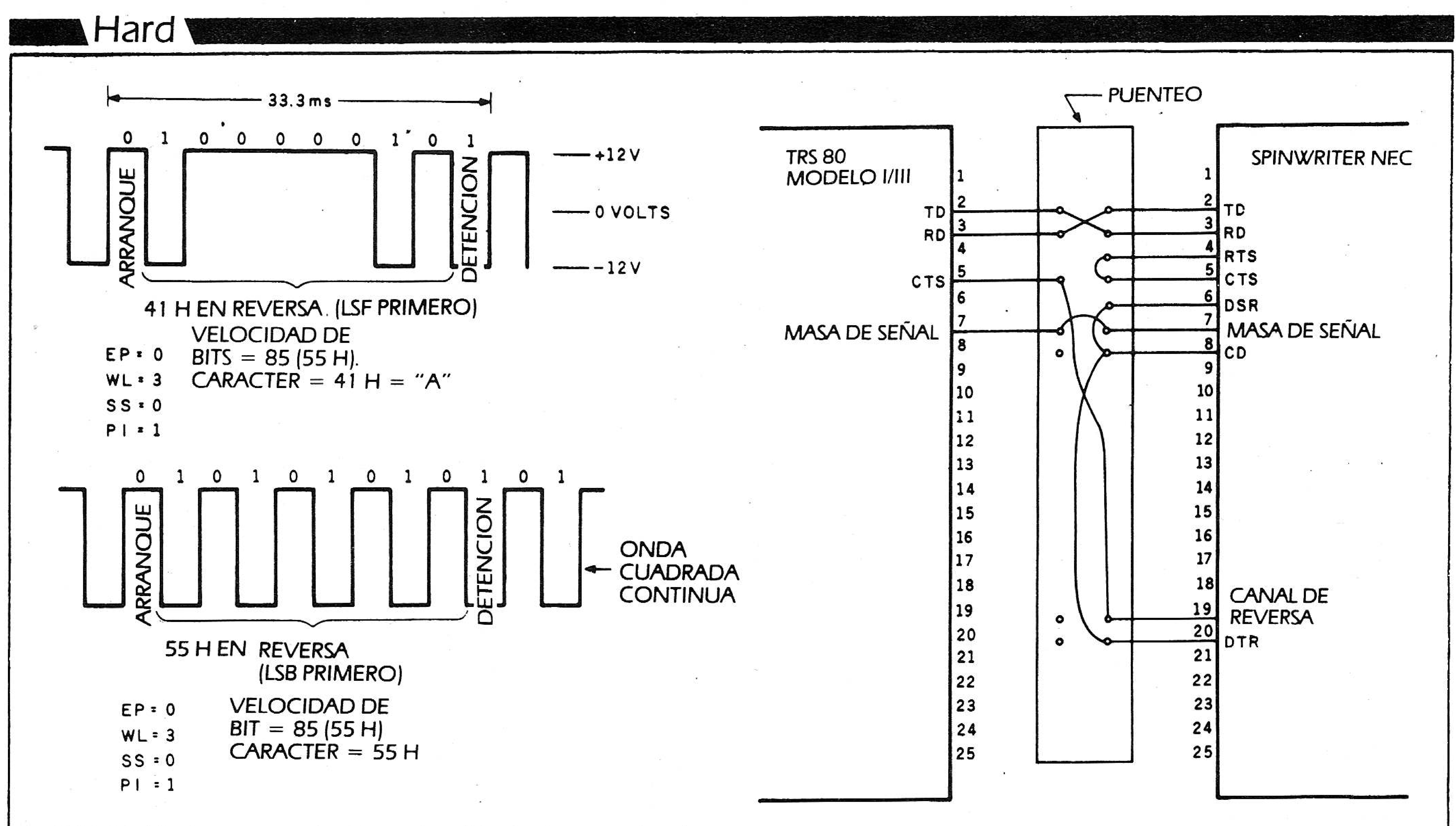


FIGURA 14: El programa ejercitador de salida de datos produce la salida TD que se puede examinar con un osciloscopio. Se puede leer fácilmente el flujo de bits en serie. La salida de un caracter 55 hexadecimal produce una onda cuadrada continua debido a los 1 y 0 alternados.

FIGURA 15: Este punteo típico RS-232C conecta una spinwriter NEC al Modelo I/III. Se deben "simular" varias líneas de Spinwriters para hacerle creer a ésta que ha recibido la señal adecuada. La línea de "canal de reversa" permite una operación de 1200 bits.

...Y SEGUIMOS CRECIENDO PROGRAMAS EN BASIC PARA INGENIEROS Y CIENTIFICOS

11 EXCELENTES CAPITULOS EN LOS QUE se desarrollan en plenitud los programas que permitirán a profesionales de la ciencia y la ingeniería resolver con precisión y velocidad, complejas operaciones matemáticas e intrincadas fórmulas físicas, químicas, electrónicas, etc.

11 EXCELENTES CAPITULOS con un nutrido temario dentro de cada uno de ellos que se publican en el siguiente orden:

Capítulo 1: Evaluación del Intérprete y Compilador Basic
Identifica puntos débiles en diversos programas comerciales Basic y suministra programas para probar cualquier Basic. Los resultados serán usados para seleccionar varias constantes y operaciones en capítulos venideros. También, en este capítulo, están las discusiones del común "NEXT sin FOR" y el problema "10 GOSUB 10".

Capítulo 2: Desviaciones Standard e Intermedias
Examina algunos algoritmos estadísticos y presenta un programa para implementarlos. También se dan rutinas para generar y probar programas semejantes y números indefinidos Gaussian.

Capítulo 3: Operaciones de Vector y Matriz
Suma las operaciones aritméticas de vector y matriz, incluyendo producto aumentado, y multiplicación e inversión de matriz. Dos importantes programas son desarrollados: uno para llevar a cabo una multiplicación y otro para calcular.

Capítulo 4: Solución Simultánea de Ecuaciones Lineales
Presenta programas para llevar a cabo los algoritmos del método Cramer, la eliminación del método Gauss, la eliminación del método Gauss-Jordan y el método Gauss Seidel, todos para resolver ecuaciones simultáneas. Se estudia además condicionamientos inferiores observando un programa que genera matrices Hilbert, y se desarrolla un programa para resolver ecuaciones con coeficientes complejos.

Capítulo 5: Desarrollo de un programa de formación de curvas
Es el primero de una serie de capítulos sobre la formación de curvas. Es una buena ilustración del programa desarrollado de punto inferior, se escribe y discute un programa lineal de formación de curvas. El programa incluye rutinas para simular datos, trazar curvas, computar la formación de curvas y completar el coeficiente de relación.

Capítulo 6: Clasificación
Describe y compara varias rutinas de clasificación Basic, incluyendo dos clases, de apariencia falsa, una del tipo "cáscara" y la otra de tipo veloz no corriente. Un tipo de rutina es incorporada en un programa de formación de curva del Capítulo 5 habilitando al programa para manejar datos experimentales reales.

Capítulo 7: Formación de curvas. Aplicaciones
Extiende al programa de Formación de Curvas a ecuaciones de polinomios y halla la Formación de Curvas por 3 ejemplos: las ecuaciones por capacidad para arder, presión de vapor, y propiedades de exceso de vapor.

Capítulo 8: Solución de Ecuaciones por el Método Newton
Presenta una serie de programas que emplean los algoritmos de Newton para encontrar las raíces de una ecuación. Esta herramienta, será usada nuevamente en el Capítulo 10 para la formación de curvas no lineal.

Capítulo 9: Integración Numérica
Desarrolla programas por tres métodos diferentes de reintegro: el trapezoidal, el de Simpson y el de Romberg. La reforma final es también examinada. El método de Simpson será usado en el Capítulo II para evaluar los errores de función de Gaussian.

Capítulo 10: Ecuaciones de Formación de Curvas no Lineales
Examina los algoritmos de la formación de curvas para la función racional y exponencial. Se dan dos ejemplos: el factor Clausing y la ecuación de difusión.

Capítulo 11: Aplicaciones avanzadas. La Curva Normal, la Función Gamma, y las Funciones Bessel
Recurre a varios temas avanzados en programación por aplicaciones matemáticas. Este último capítulo suma y expande sobre un número de conceptos presentados en este libro.

Cada capítulo contiene, además, ejercicios destinados a extender la comprensión del material al lector.

Para los lectores que se están acercando al Basic por primera vez, se incluye en los apéndices, un resumen de la sintaxis, funciones standard, y palabras reservadas al Basic. La experiencia educacional real de este libro será, sin embargo, adquirida con trabajo cuidadoso a través de los mismos programas.

Para la obtención del sumario completo marque el nro. 99 en el talón de Servicios al Lector.

MARQUE CON UN CIRCULO EL 76 EN LA SERVICE CARD



KILOMETROS

Objetivo: Calcular kilómetros (km) a partir de millas náuticas.

Variables necesarias:
N (nmi).

Variables transformadas:
Ninguna.

Variables obtenidas:
K (km).

Ecuación: km = 1,852 nmi.

Listado de la subrutina:

```
 4REM KILOMETROS
10 LET K = 1,852*N
20 RETURN
```

Prueba para kilómetros/millas náuticas

```
RUN
ENTRADA DE LAS VARIABLES
?5
?10
?15
?20
?25
?30
?35
?40
?80
?160
?320
RESULTADOS:
SALIDA
```

ENTRADA	SALIDA
5	9.2600000
10	18.520000
15	27.780000
20	37.040000
25	46.300000
30	55.560000
35	64.820000
40	74.080000
80	148.16000
160	296.32000
320	592.64000

FINALIZADO

KILOMETROS

Objetivo: Calcular kilómetros (km) a partir de millas terrestres.

Variables necesarias:
M (mi).

Variables transformadas:
Ninguna.

Variables obtenidas:
K (km).

Ecuación: km = 1,609 mi.

Listado de la subrutina:

```
 4REM KILOMETROS
10 LET K = 1,609*M
20 RETURN
```

Prueba para kilómetros/millas terrestres

```
RUN
ENTRADA DE LAS VARIABLES
?5
?10
?15
?20
?25
?30
?35
?40
?80
?160
?320
RESULTADOS:
SALIDA
```

ENTRADA	SALIDA
5	8.0467200
10	16.093440
15	24.140160
20	32.166880

25	40.233600
30	48.280320
35	56.327040
40	64.373760
80	128.74752
160	257.49504
320	514.99008

FINALIZADO

LITROS

Objetivo: Calcular litros (l) a partir de galones U.S. (gal).

Variables necesarias:
G (gal).

Variables transformadas:
Ninguna.

Variables obtenidas:
L (l).

Ecuación: l = 3,785 gal.

Listado de la subrutina:

```
 4REM LITROS
10 LET L = 3,785*G
20 RETURN
```

Prueba para litros/gal. U.S.

```
RUN
ENTRADA DE LAS VARIABLES
?1
?2
?4
?8
?16
?32
?64
?128
?256
?512
?1024
RESULTADOS:
SALIDA
```

ENTRADA	SALIDA
1	3.7850000
2	7.5700000
4	15.140000
8	30.280000
16	60.560000
32	121.12000
64	242.24000
128	484.48000
256	968.96000
512	1937.9200
1024	3875.8400

FINALIZADO

METROS

Objetivo: Calcular metros (m) a partir de yardas (yd).

Variables necesarias:
Y (yd).

Variables transformadas:
Ninguna.

Variables obtenidas:
M (m).

Ecuación: m = 0.9144 yd.

Listado de la subrutina:

```
 5REM METROS
10 LET M = 0.9144*Y
20 RETURN
```

Prueba para metros

```
RUN
ENTRADA DE LAS VARIABLES
?1
?10
?100
?1000
?1760
?50
?25
?5
?12.5
?2
?3.25
RESULTADOS:
SALIDA
```

ENTRADA	SALIDA
1	.91440000
10	9.1440000
100	91.440000
1000	914.40000
1760	1609.3440
50	45.720000
25	22.860000
5	4.5720000
12.5	11.430000
2	1.8288000
3.25	2.9718000

FINALIZADO

METROS POR SEGUNDO

Objetivo: Calcular metros por segundo (m/s) a partir de kilómetros por hora (km/hr).

Variables necesarias:
K (km/hr).

Variables transformadas:
Ninguna.

Variables obtenidas: M (m/s).

Ecuación: m/s = 0.2778 km/hr.

Listado de la subrutina:

```
 5REM METROS/SEGUNDO
10 LET M = 0.2778*K
20 RETURN
```

Prueba para metros/segundo (KM)

```
RUN
ENTRADA DE LAS VARIABLES
?1
?2
?4
?8
?16
?32
?64
?128
?256
?512
RESULTADOS:
SALIDA
```

ENTRADA	SALIDA
1	0.2778000
2	0.5556000
4	1.1112000
8	2.2224000
16	4.4448000
32	8.8896000
64	17.779200
128	35.558400
256	71.116800
512	142.23360

FINALIZADO



Informe histórico de las cuentas a cobrar

VISICALC

Descripción

L habilidad del VisiCalc para mover bloques de datos específicos para almacenamiento en disco ha sido empleada en este ejemplo para desplazar valores de una zona de trabajo para su re-entrada en otras zonas de la hoja de trabajo para referenciado y uso en fórmulas.

Para demostrar esta habilidad/capacidad, se ha preparado un mayor Informe Histórico de Cuentas a Cobrar. Para analizar la antigüedad de las cuentas enumeradas, una vez por mes se realiza una operación de actualización. Las cuentas corrientes y aquéllas con una antigüedad mayor de 90 días, conjuntamente con una columna en blanco inmediatamente a su izquierda, se llevan a un disco de almacenamiento, luego son reingresadas en la hoja mayor, y reposicionada una columna a la derecha. Los valores por sobre los 60 y 90 días se llevan a un disco de almacenamiento, luego son reingresados en un AREA DE TRABAJO (WORK AREA) para una función de acumulación.

Operaciones realizadas

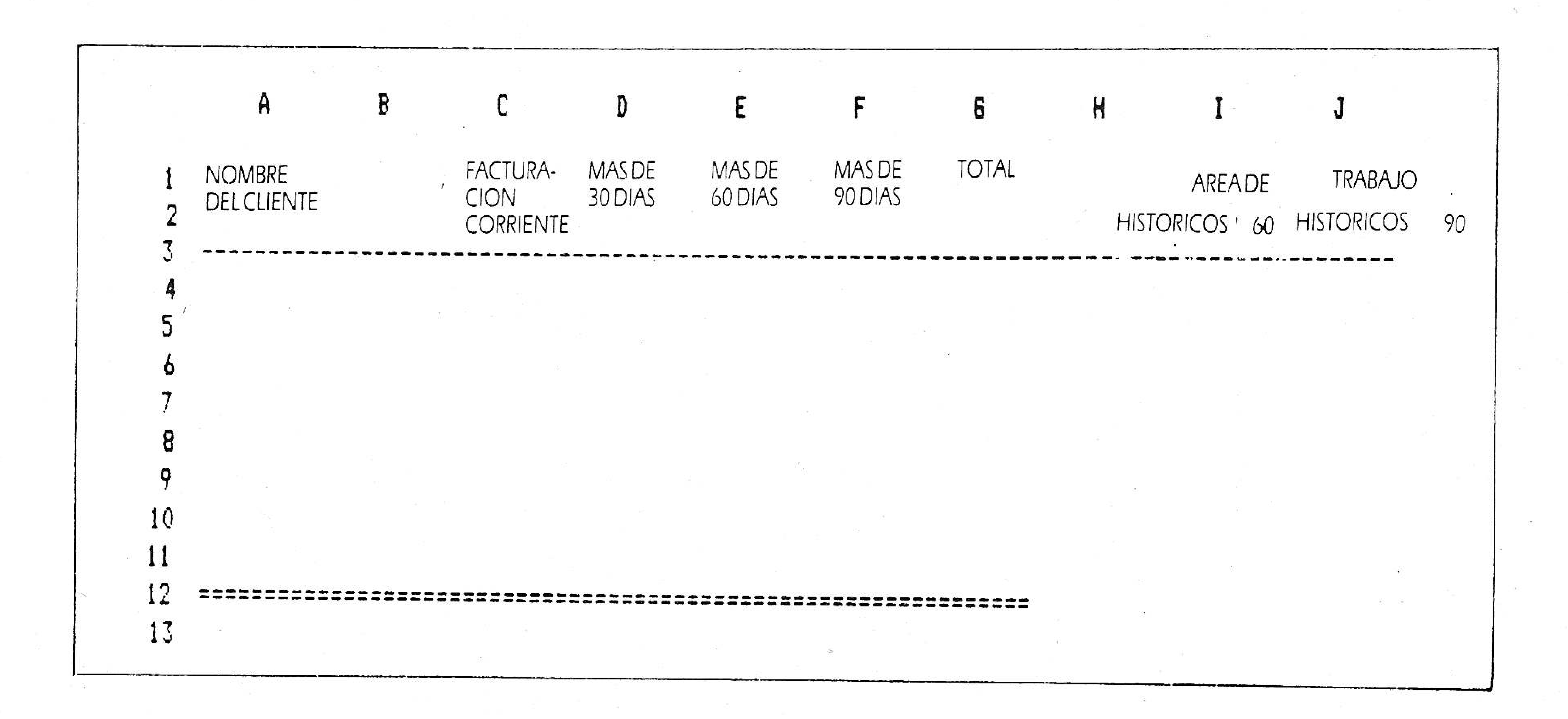
Preparación del Formato.
Ingreso de fórmulas matemáticas.
Realización de entradas al libro mayor.
Actualización del libro mayor.
Realización de entradas mensuales.
Realización de entradas adicionales.
Almacenamiento.
Impresión.

Funciones utilizadas

SUM (suma)

Ordenes utilizadas

REPEAT LABEL (repetir identificación)	
FORMAT (formato)	R = alineación del margen a la derecha
GLOBAL (global)	$ = formato de pesos y centavos
STORAGE (almacenamiento)	# = almacena un archivo de formato intercambio de datos.
REPLICATE (repetir)	copia
INSERT (insertar)	R = fila
DELETE (borrar)	R = fila



Preparación del formato

Utilizando las siguientes direcciones, prepare su hoja de diario mayor copiando exactamente la figura 1 en la forma en que se ilustra, conservando las ubicaciones de filas y columnas en forma exacta para toda la información.

Para dar formato a todas las ubicaciones para que se exhiban las entradas de valores en pesos y centavos, tipee:

/G	pone en funcionamiento la orden GLOBAL
F	FORMAT
$	exhibe pesos y centavos

Para colocar los encabezados de columna, coloque el cursor donde usted desee realizar la entrada y tipee:

/F	pone en funcionamiento la orden FORMAT
R	Alineación del margen a la derecha

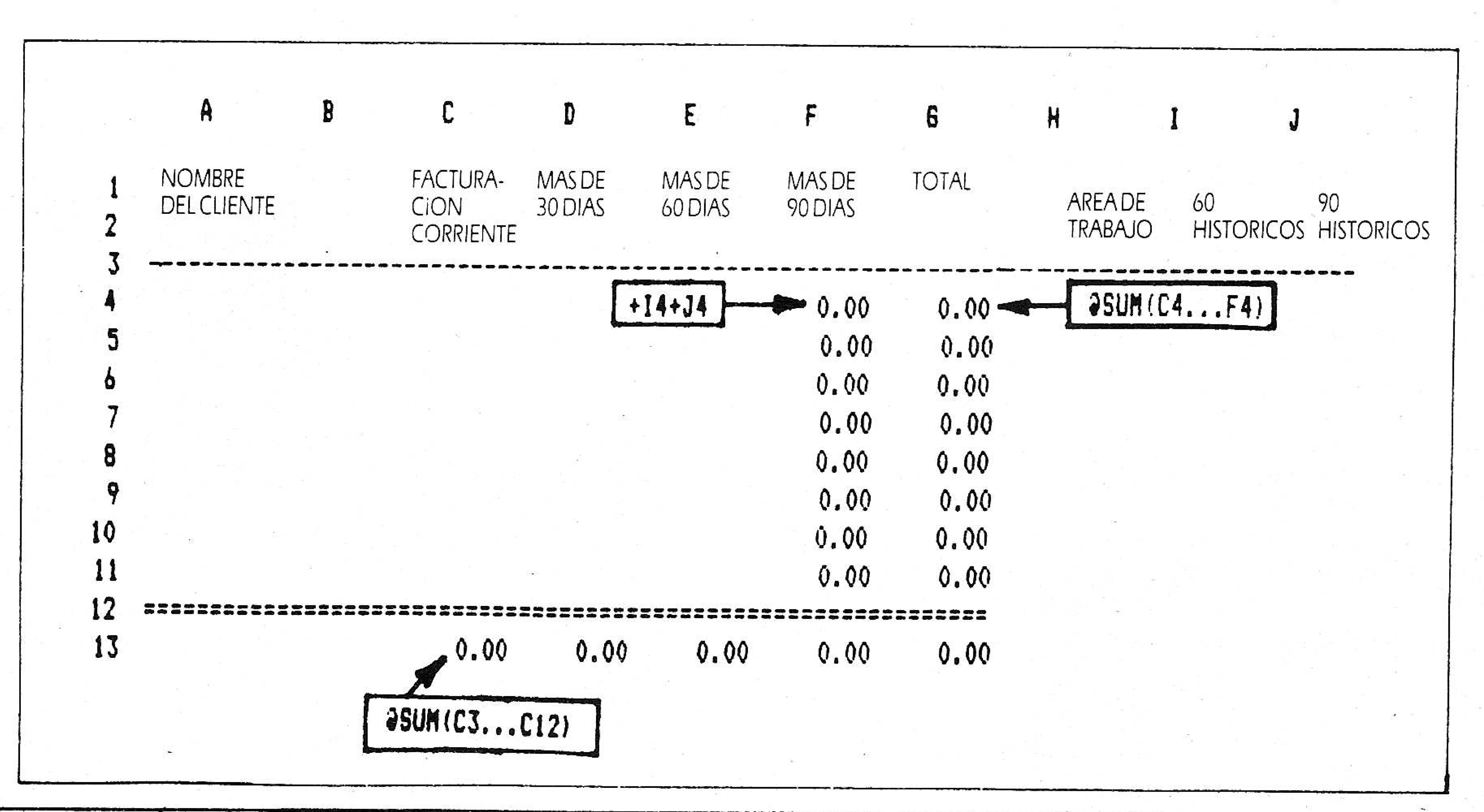
Tipee el título de la columna. Pulse la tecla cursor (flecha) para pasar a su siguiente ubicación.

Pulsando la tecla cursor en esta operación, tanto ingresa el título de la columna dentro de la ubicación y mueve automáticamente su cursor a la siguiente ubicación para realizar el tipeo. Tipee el resto de encabezamientos de su columna utilizando la secuencia de órdenes mencionada anteriormente.

Para ingresar las líneas discontinuas en su hoja mayor, coloque su cursor en la columna que se encuentra más a la izquierda que corresponda a la fila donde usted desea la línea (línea A3 en este ejemplo). Tipee:

/–	pone en funcionamiento la orden REPEAT LABEL
–	identificación a repetirse
RETURN	ejecuta la orden

La columna en la que se encuentra el cursor tendrá ahora una línea de guiones (–) en todo su ancho. Para prolongar esta línea discontinua en la misma fila para las restantes co-



lumnas, deje el cursor donde está y tipee:

/R	pone en funcionamiento la orden REPLICATE
RETURN	le dice a la orden que copie la línea de guiones en la que se encuentra el cursor.
B3	primera coordenada en la fila desde donde usted desea que se prolongue la línea discontinua
	salto... desde - hasta
J3	última coordenada en la fila en la que usted desea que se prolongue la línea discontinua
RETURN	ejecuta la orden.

La línea discontinua ahora aparecerá prolongada en todas las columnas que usted ha indicado mediante las coordenadas. Para ingresar una doble línea discontinua (= = =) en su hoja mayor, repita las operaciones indicadas anteriormente utilizando el símbolo = como la identificación a repetirse.

Ingreso de fórmulas matemáticas

Usted ahora comenzará a ingresar fórmulas matemáticas que establecerán las relaciones entre las posiciones de columnas y filas. Las fármulas y sus ubicaciones se ilustran en la figura 2.

La primera fórmula agregará los valores en la columna FACTURACION CORRIENTE.

Coloque su cursor en C13 y tipee:

@SUM	agrega los valores en la lista.
C3	primera coordenada de la columna que usted desea agregar.
	·indica desde - hasta
C12	última coordenada de la columna que usted desea agregar
RETURN	ingresa la fórmula.

Su siguiente operación es copiar la fórmula recién ingresada en la parte inferior de cada columna que usted desea agregar.

Deje el cursor en C13 y tipee:

/R	pone en funcionamiento la orden REPLICATE.
RETURN	le dice a la orden que copie la fórmula en C13.
D13	primera coordenada donde usted desea copiar la fórmula en las columnas.
	indica desde - hasta.
G13	última coordenada donde usted desea copiar la fórmula en las columnas.
RETURN	ejecuta la orden y prepara para recibir instrucciones adicionales.
R	le dice a la orden que copie la dirección de la coordenada en la fórmula relativa a su nueva ubicación.

La tercera fórmula agregará los valores en las dos columnas de AREA DE TRABAJO (WORK AREA) y exhibirá la respuesta en la columna **más de 90 días**. Este valor reflejará el valor acumulado de las cuentas a cobrar tomadas más de 90 días.

Coloque su cursor en F4 y tipee:

+	prepara la coordenada para aceptar una expresión numérica.
I4	coordenada que contiene el valor.
+	suma
J4	coordenada que contiene el valor.
RETURN	ingresa la fórmula.

Luego, ingrese la fórmula en la columna TOTAL para agregar la suma (SUM) de los valores en cada columna en la fila a la izquierda.

Coloque su cursor en G4 y tipee:

@SUM	agrega los valores en la lista.
C4	primera coordenada de la fila que usted desea agregar.
F4	última coordenada de la fila que usted desea agregar.
RETURN	ingresa la fórmula.

Ahora será necesario copiar las dos fórmulas recientemente ingresadas en cada fila en sus columnas correspondientes (MAS DE 90 DIAS y TOTAL).

Coloque su cursor en F4 y tipee:

/R	pone en funcionamiento la orden REPLICATE
G4	copia todas las entradas a través de las columnas F4 a G4.
RETURN	prepara para recibir información adicional.
F5	primera coordenada donde desea copiar las fórmulas debajo de las columnas.
	elipsis... indica desde - hasta.
F11	última coordenada donde usted desea copiar las fórmulas debajo de las columnas.
RETURN	ejecuta la orden y prepara para recibir instrucciones adicionales.
R	le dice a la orden que copie la dirección de
R	la coordenada en la fórmula relativa a su
R	nueva ubicación.

Realizando entradas al mayor

Su mayor histórico de cuentas a cobrar ahora ya está preparado. Una vez por mes todo lo que tiene que realizar es llevar a cabo el proceso de actualización que se describe en la siguiente sección, y realizar las entradas de facturación corriente. Para realizar la siguiente serie de ejercicios, tipee las entradas ilustradas en la figura 3. Para este ejemplo, las entradas han sido seleccionadas para ilustrar un mayor en operación más de 90 días.

Notas

No tipee en la columna MAS DE 90 DIAS. El valor a indicarse en la columna MAS DE 90 DIAS deberá ser tipeado en la fila adyacente de la columna 90 ANTERIORES en el AREA DE TRABAJO. Quedará exhibido en la columna MAS DE 90 DIAS mediante la fórmula allí ingresada.

Nunca ingrese valores en coordenadas que contengan fórmulas dado que de realizarse así se borrarán las fórmulas.

La Columna B debe permanecer en blanco para este ejemplo.

	A	B	C	D	E	F	G	H	I	J
1 2	NOMBRE DEL COMPRADOR		FACTURACION CORRIENTE	MAS DE 30 DIAS	MAS DE 60 DIAS	MAS DE 90 DIAS	TOTAL	AREA DE TRABAJO	60 HISTORICOS	90 HISTORICOS
3	----------									
4	ACME CO.			45.00		0.00	45.00			
5	BELL CO.				25.00	45.00	70.00			45.00
6	KOLL CO.			56.58		0.00	56.58			
7	MAXEL CO.					89.00	89.00			89.00
8	REDDY CO.				35.00	0.00	35.00			
9	AJAX CO.		75.16			15.00	90.16			15.00
10	ZIPLOK		84.00			0.00	84.00			
11	MULTI-CR		3578.00			0.00	3578.00			
12	==========									
13			3737.16	101.58	60.00	149.00	4047.74			

Actualización del mayor

Para llevar a cabo el proceso de actualización, usted transferirá los valores en la Columna B (en blanco) y las columnas FACTURACION CORRIENTE y MAS DE 30 DIAS dentro de un archivo de almacenamiento en disco. Entonces usted moverá los valores en las columnas MAS DE 60 DIAS y MAS DE 90 DIAS a un archivo de almacenamiento en disco separado. En el tercer paso, usted volverá a ingresar los valores en la columna B (en blanco) y las columnas FACTURACION CORRIENTE y MAS DE 30 DIAS se reposicionarán una columna hacia la derecha. Esto mueve cada uno de los valores hacia la derecha dentro de su nueva columna histórica, y borra la columna de FACTURACION CORRIENTE.

El paso final en el proceso de actualización vuelve a ingresar los valores de las columnas MAS DE 60 DIAS y MAS DE 90 DIAS en las columnas del AREA DE TRABAJO 60 HISTORICOS y 90 HISTORICOS. La fórmula en la columna MAS DE 90 DIAS agrega las sumas en cada fila de estas dos columnas y exhibe los resultados en la columna MAS DE 90 DIAS como los totales acumulados para cada cliente indicado en la lista.

Coloque su cursor en B4 (la coordenada superior izquierda del área de la hoja mayor rectangular que usted desea copiar en el archivo almacenado).

Tipee:

/S	pone en funcionamiento la orden STORAGE
#	almacena un archivo de Formato de Intercambio de Datos (DIF)
S	almacena
SIXTYDAY	nombre del archivo; no tipee espacios entre palabras
RETURN	prepara para recibir información adicional.
D11	coordenada inferior derecha del rectángulo de entradas de valor a ser almacenadas.
RETURN	prepara para recibir instrucciones adicionales.
C	almacena los valores en el formato de la columna y ejecuta la orden.

Coloque su cursor en E4 (la coordenada superior izquierda del área de la hoja del mayor rectangular que usted desea copiar en el archivo almacenado) y tipee:

/S	pone en funcionamiento la orden STORAGE
#	almacena un archivo de Formato de Intercambio de Datos (DIF)
S	almacena
NINETYDAY	nombre del archivo; no tipee espacios entre palabras
RETURN	prepara para recibir información adicional.
F11	coordenada inferior derecha del rectángulo de entradas de valor a almacenarse.
RETURN	prepara para recibir instrucciones adicionales.
C	almacena los valores en el formato de la columna y ejecuta la orden.

El tercer paso en la operación de actualización vuelve a ingresar los valores del archivo SIXTYDAY sobre la hoja del mayor una columna a la derecha.

Coloque su cursor en C4 (la coordenada superior izquierda del área de la hoja del mayor rectangular donde usted desea que se vuelvan a reingresar los valores) y tipee:

/S	pone en funcionamiento la orden STORAGE
#	carga un archivo de Formato de Intercambio de Información (DIF)
L	carga
SIXTYDAY	nombre del archivo; no tipee espacios entre palabras
RETURN	prepara para recibir instrucciones adicionales.
C	carga los valores en el formato de la columna y ejecuta la orden.

La operación final ingresa kis vakires del archivo NINETYDAY dentro de las columnas del AREA DE TRABAJO.

Coloque su cursor en I4 (la coordenada superior izquierda del área de la hoja del mayor rectangular donde usted desea que se vuelvan a ingresar los valores). Tipee:

	A	B	C	D	E	F	G	H	I	J
1 2	NOMBRE DEL CLIENTE		FACTURACION CORRIENTE	MAS DE 30 DIAS	MAS DE 60 DIAS	MAS DE 90 DIAS	TOTAL	AREA DE TRABAJO	60 HISTORICOS	90 HISTORICOS
3	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------
4	ACME CO.				45.00	0.00	45.00			0.00
5	BELL CO.					70.00	70.00		25.00	45.00
6	KOLL CO.				56.58	0.00	56.58			0.00
7	MAXEL CO.					89.00	89.00			89.00
8	REDDY CO.					35.00	35.00		35.00	0.00
9	AJAX CO.			75.16		15.00	90.16			15.00
10	ZIPLOK			84.00		0.00	84.00			0.00
11	MULTI-CR			3578.00		0.00	3578.00			0.00
12	============	============	============	============	============	============	============			
13			0.00	3737.16	101.58	209.00	4047.74			

/S — pone en funcionamiento la orden STORAGE
— carga un archivo con Formato de Intercambio de Datos (DIF)
L — carga
NINETYDAY — nombre del archivo; no tipee espacios entre palabras
RETURN — prepara para recibir instrucciones adicionales.
C — carga los valores en el formato de la columna y ejecuta la orden.

Ahora usted ha completado su actualización mensual de las entradas existentes. Su mayor se verá como el indicado en la Figura 4. Usted está listo para ingresar las transacciones que ha acumulado durante el mes que pasó recientemente.

Realización de entradas mensuales

Las entradas mensuales al mayor tomarán una de las siguientes dos formas: pagos y facturación corriente.

Para realizar las entradas de la facturación corriente, tipéelas directamente en la columna de FACTURACION CORRIENTE.

Para realizar una entrada por pago en las columnas MAS DE 30 DIAS y MAS DE 60 DIAS, coloque su cursor sobre el valor que desea restar y tipee:

— prepara para usar el valor
– — resta

Tipee el valor de pago:

RETURN — ingresa el valor

Para realizar una entrada por pago en la columna MAS DE 90 DIAS, coloque su cursor en la fila adyacente en la columna de AREA DE TRABAJO que contiene un valor y tipee:

— prepara para usar el valor
– — resta

Tipee el valor de pago:

RETURN — ingresa el valor

Realización de entradas adicionales

Para agregar entradas, usted tendrá que adicionar nuevas filas. Estas nuevas entradas se pueden realizar al final de la lista existente, o en orden alfabético. Todas las funciones SUM que agregan totales de columna se ajustarán automáticamente a los efectos de incluir las nuevas filas en la medida que usted coloque las filas entre las coordenadas en la fórmula original. Las fórmulas que realizan otras funciones dentro de las columnas expandidas, no obstante deberán ser ingresadas dentro de las nuevas coordenadas de entrada en cada columna donde se utilice una fórmula. Estas fórmulas existentes pueden copiarse en las nuevas coordenadas en forma individual o utilizando la orden REPLICATE.

Para colocar una nueva fila, coloque su cursor en la fila que usted desea mover hacia abajo y quedará colocada una fila en blanco.

Usted ahora puede comenzar a ingresar fórmulas donde sea necesario, luego comience a realizar sus nuevas entradas.

Almacenamiento

En algunos casos usted puede desear almacenar su formato de trabajo o trabajo completado en un archivo en disco para una búsqueda posterior.

Para almacenar o guardar la hoja de trabajo en su totalidad, tipee:

/S — pone en funcionamiento la orden STORAGE
S — almacena (guarda)
FILENAME — nombre del archivo; no tipee espacios entre palabras
RETURN — ejecuta la orden.

Impresión

Usted puede desear imprimir una parte o la totalidad de su hoja de trabajo para archivo o distribución.

Coloque su cursor en la coordenada superior izquierda del rectángulo del área de la hoja de trabajo que usted desea imprimir y tipee:

/P — pone en funcionamiento la orden PRINT
P — impresora

Tipee la dirección de la coordenada interior izquierda del rectángulo del área de la hoja de trabajo que usted desea copiar y tipee:

RETURN — ejecuta la orden.

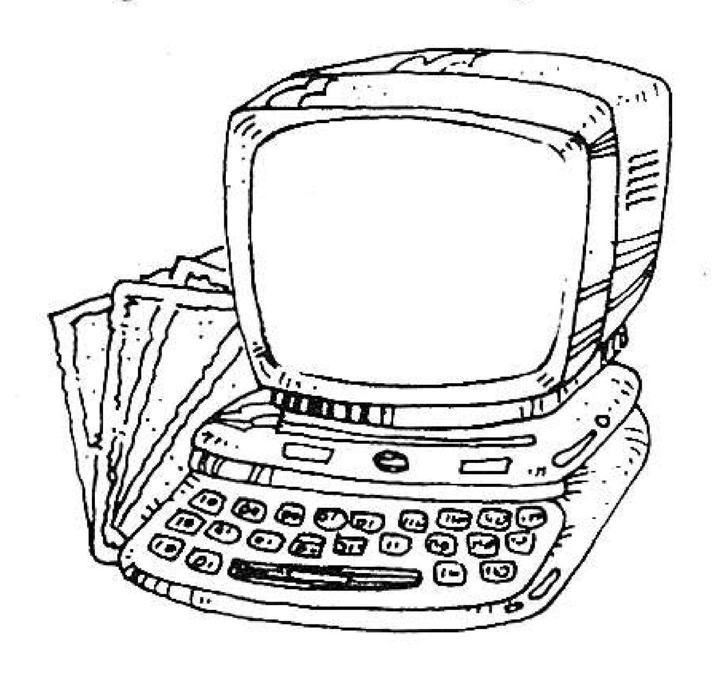
Nuevos Productos

MARQUE CON UN CIRCULO EL 88 EN LA SERVICE CARD

ALCANCE Y TOQUE SU PANTALLA

El Sistema de Entrada por Tacto de Carroll hace responder a una computadora de una manera tan sencilla como lo es pulsar el punto apropiado en su pantalla de video.

El sistema opera a través de una red de diodos emisores de luz infrarrojos que se montan en un lado y la parte inferior de un "marco de ventana" realizado de tarjetas de circuitos impresos. Esta disposición crea un reticulado X-Y de haces de luz invisibles que forma un punto de tacto donde existe una intersección de un haz X y un haz Y. El marco de ventana está montado directamente sobre la caja de la pantalla de video, con el reticulado paralelo a y directamente en frente de la superficie de la pantalla.



El área de la ventana está sometida constantemente a un rastreo para detectar las obstrucciones de los haces de luz que ocurren cuando usted toca la pantalla.

A diferencia de otros tipos de sistemas de toque al tacto, el sistema de Entrada por tacto Carroll no requiere una superposición sobre la pantalla de video, de manera tal que nada se refleja, distorsiona o interfiere con la imagen clara y nítida. Además, usted puede limpiar la pantalla sin temor de dañar el dispositivo de entrada por tacto.

Este sistema sirve para la mayoría de las computadoras populares, así como también para los terminales de computadora.

Solución al diseño y dibujo automatizado en la HP 1000 Modelo 6

Hwlett-Packard y Holguin & Associates Inc., ofrecen uno de los sistemas de diseño y dibujo más poderoso y económico de la industria de la computación. El sistema combina la Microcomputadora HP 1000 Modelo 6 de alta performance y bajo costo y el software de Dibujo automatizado CEADS-CADD de Holguin & Associates.

Esta combinación, Modelo 6/ Sistema CEADS-CADD puede utilizarse en una amplia gama de aplicaciones de diseño y dibujo; desde el dibujo mecánico, diseño de arquitectura y diagramas esquemáticos detallados hasta diseño de plaquetas de circuito impreso.

La productividad de dibujo se puede aumentar de un 25 a un 50%, con revisión y edición instantánea, y la capacidad de trazar un plano con sólo tocar el teclado o una tableta gráfica. El plazo típico de capacitación es de sólo unos pocos días. La interconexión cómoda del usuario combinado con un manual simple y comprensible elimina la necesidad de experiencia.

El sistema se dispone en la forma de sistemas listos para usar y sistemas adaptables. Ambas versiones están disponibles como sistema integrado único (un terminal operando CEADS-CADD) o un sistema múltiple..

Además, la novísima microcomputadora de Hewlett-Packard Modelo 6 aporta para este sistema de dibujo y diseño, características que le son propias, como un conjunto operable sobre la mesa, incluyendo:

* multiprogramación en tiempo real.
* lenguajes de programas de alto nivel.
* configuración de sistema distribuído.
* poderosa terminal de gráficos HP 2623A.
* elección de almacenamiento masivo de 16,27 ó 64 Mb, con cassette para una fácil reserva o respaldo.
* graficador multilápiz (disponible en 8 colores) capaz de manejar elementos en formatos standard A/A4 hasta D/A1.
* traducción a valores numéricos para una fácil interacción gráfica.

Otra ventaja importante de la combinación Modelo 6/CEADS-CADD es que excede la performance documentada de los sistemas CAD de vanguardia a menos de la mitad del costo!!

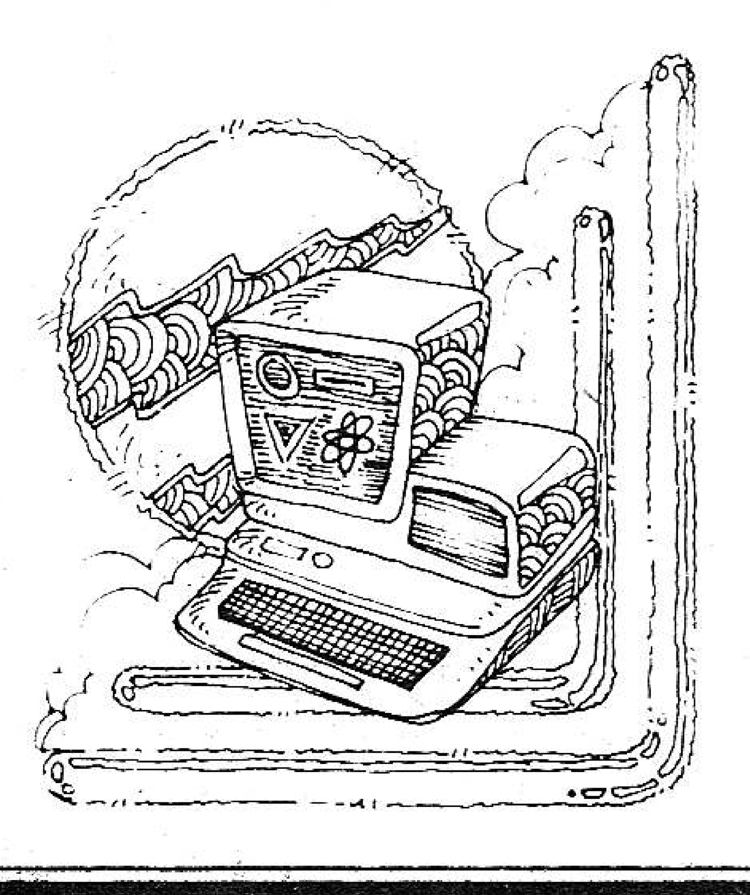


Unidad de discos rígidos para Commodore

Dos nuevas unidades discos rígidos permite a los usuarios de las microcomputadoras Commodore aumentar la capacidad de almacenamiento de sus sistemas.

Denominadas 9065D y 9150D, estas nuevas unidades se basan en la tecnología Winchester de 5 1/4". La unidad 9065D ofrece 6,4 megabytes (sin formatear) de almacenamiento mientras que 9150D es capaz de almacenar 9,6 megabytes (sin formatear). Ambas unidades contienen una interconexión IEEE-488 incorporada para acoplarse a cualquier Commodore PET, CBM o SuperPET.

Los discos rígidos son compatibles con DOS con los mecanismos impulsores de discos flexibles corrientes.



Plaqueta de entrada/salida en serie

PSIO, plaqueta de entrada/salida en serie programable, es totalmente compatible con el software corriente de Apple incluyendo Pascal 1.1, Applesoft e Integer Basic. Suministra una interconexión total EIA RS-232 DTE para la mayoría de los modems, impresoras y terminales TRC.

El software ROM de 1K de la plaqueta permite al usuario un control mayor de la impresora programable, retardo luego de retorno de carro, generación automática de alimentación de línea, exhibición de video. El formato es alterable por el usuario con las configuraciones más populares tales como las condiciones de defecto por error.

ADAPTANDO AL CP/M CON EL MODELO III

Los usuarios del Modelo III TRS-80 pueden ahora ingresar al software de base CP/M gracias al Shuffleboard III (plaqueta de adaptación), una plaqueta enchufable provista por Memory Merchant. La Shuffleboard III es el primer y más avanzado sistema CP/M 2,2 para el Modelo III, permitiendo a los usuarios operar programas populares tales como Wordstar, SuperCalc y Mailmerge. La plaqueta incluye 16 Kbytes de memoria de acceso directo (RAM) dándole al Modelo III la potencia de un CP/M de 64 Kbytes sin interferir con la memoria sólo para leer (ROM) o la memoria de video. Shuffleboard III es totalmente compatible con cualquier sistema operativo en disco (DOS) de TRS-80 incluyendo TRSDOS, LDOS y NEWDOS.

Instale la plaqueta enchufándola en los dos encastres existentes dentro del Modelo III.

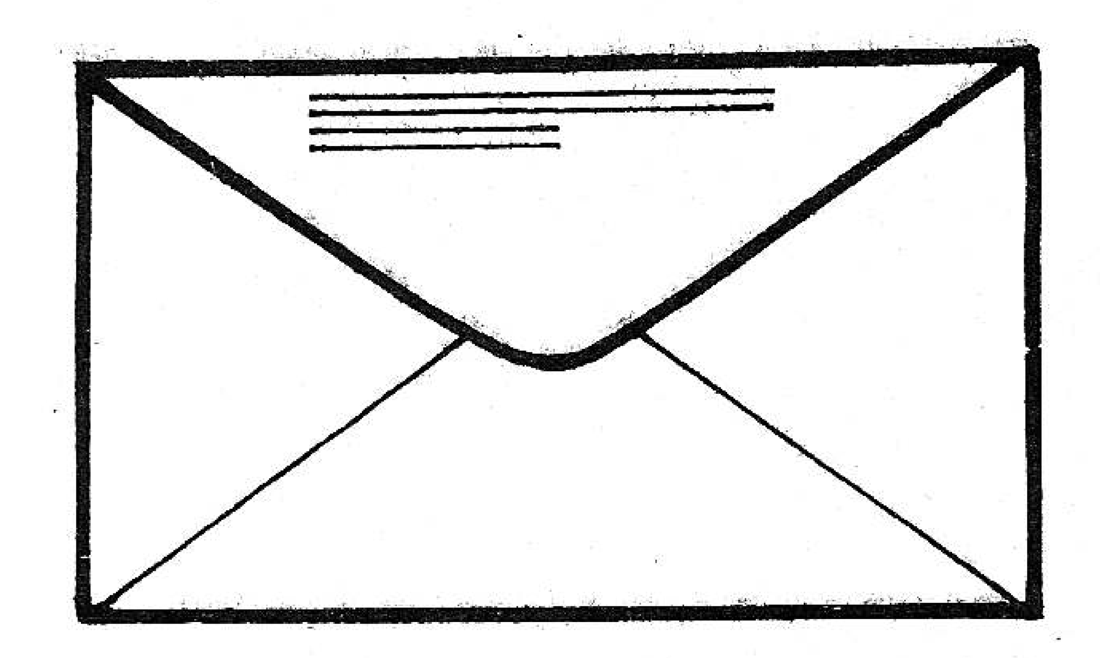
Buzón

Señorita:
Graciela R. Aliguer

Lamentamos la demora en cursar esta respuesta a su requerimiento pero de cualquier manera cumplimos en informarle que el CENTRO ESPECIALIZADO PARA LA ENSEÑANZA DE LAS CIENCIAS tiene su sede en la calle Cangallo 1719 Piso 1ro. Dto. "A" y que los próximos cursos de programación de microcomputadoras en lenguaje BASIC se inician el 9 de Mayo. Con respecto a las vacantes le sugerimos se documente en el establecimiento, dado que la diferencia de tiempo entre nuestros datos y la fecha de salida de MICROCOMPUTACION puede inducir a errores.

Señor:
José Alberto Ricciardi

Agradecemos los elogios como así también la sugerencia y paso a contestarle los cinco puntos de su carta.
1.– Es muy difícil hacer una comparación directamente en las distintas versiones de BASIC sin caer en omisiones, por lo tanto le sugerimos nos diga qué programa quiere correr en su TI 99/4 y a vuelta de correo recibirá las modificaciones que deben realizarse.
2.– Muy buena la sugerencia de la ficha técnica de Software. Ya mismo la cursaremos al Departamento de Producción.
3.– Tenemos en estudio la reimpresión de los números agotados.
4.– Nos gustaría que se explayase mejor respecto a esta sugerencia.
5.– Hay organismos especializados que nuclean Institutos de Enseñanza privados u oficiales donde se puede recabar información respecto a títulos y duración de carreras.



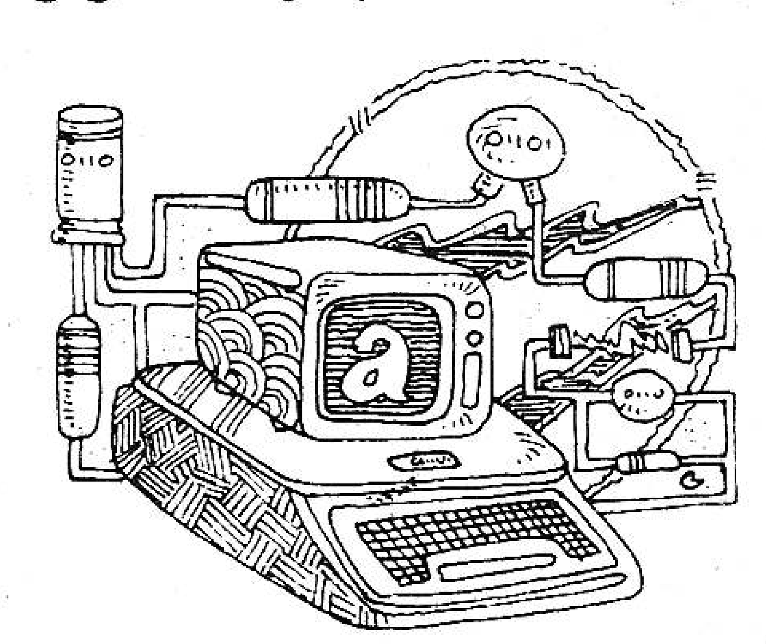


The thinking computer. Mind inside matter. Bertram Raphael

W.H. Freeman and Company

Este excelente libro está perfectamente adecuado para los lectores inclinados a los temas técnicos que desean conocer más acerca de la inteligencia artificial y los robots. Escrito por uno de los pioneros en la investigación de la inteligencia artificial, este libro suministra una cobertura comprensiva y actualizada de este campo en un estilo que balancea la profundidad técnica con la facilidad de lectura y comprensión. Raphael tiene una inclinación especial para ilustrar ideas abstractas con ejemplos memorables, como un crucigrama "criptoaritmético" que se realiza mediante una exploración o búsqueda ramificada, y un misterio de mayordomo y ama de llaves que se soluciona mediante técnicas de exploración mediante teoremas.

La introducción del libro que suministra una orientación básica sobre computación para los recién iniciados en este campo, también versa sobre periféricos y software de inteligencia artificial como así también sondea brevemente sobre la mala interpretación de las computadoras en dos puntos: que son solamente calculadoras aritméticas gigantes y que son bobas.



A collection of programming problems and techniques

Una de las cosas que suelen gustar en el campo de la computación es que inspira un nuevo estilo de soledad y que siendo persistente se puede ganar siempre. Un problema frecuente es encontrar el juego que tanto provoque satisfacción y que agudice los sentidos y nuestra habilidad. Muchos libros de programación para principiantes sólo dan ejemplos modestos e indican problemas que no ponen al estudiante en posición de desafío.

Para aquéllos que desean encontrarse frente a frente con el desafío mismo, este libro provee casi 400 problemas de diversa dificultad y con diferentes niveles. Estos ejercicios suministran la experiencia que normalmente suele presentarse a los programadores. Una visión interna de los misterios de la matemática aplicada a alto nivel aún cuando no se tiene el suficiente nivel, por cuanto el nivel educacional no lo requiere.

Algunos capítulos versan sobre cómo realizar un calendario, también se describen leyendas de la historia a las que usted tiene la posibilidad de acceder y quizás develar el misterio. Hay una excelente sección sobre los lenguajes comunes de programación, así como problemas sobre órbitas de satélites.

También encontramos información sobre compiladores y graficado tridimensional, clasificación y fusión de datos.

Lo importante de este libro es que le promete al lector convertirse en un programador "a prueba de todo", que es a lo que todos aspiramos.

Curso de programación en Basic para todos

CAPITULO VI

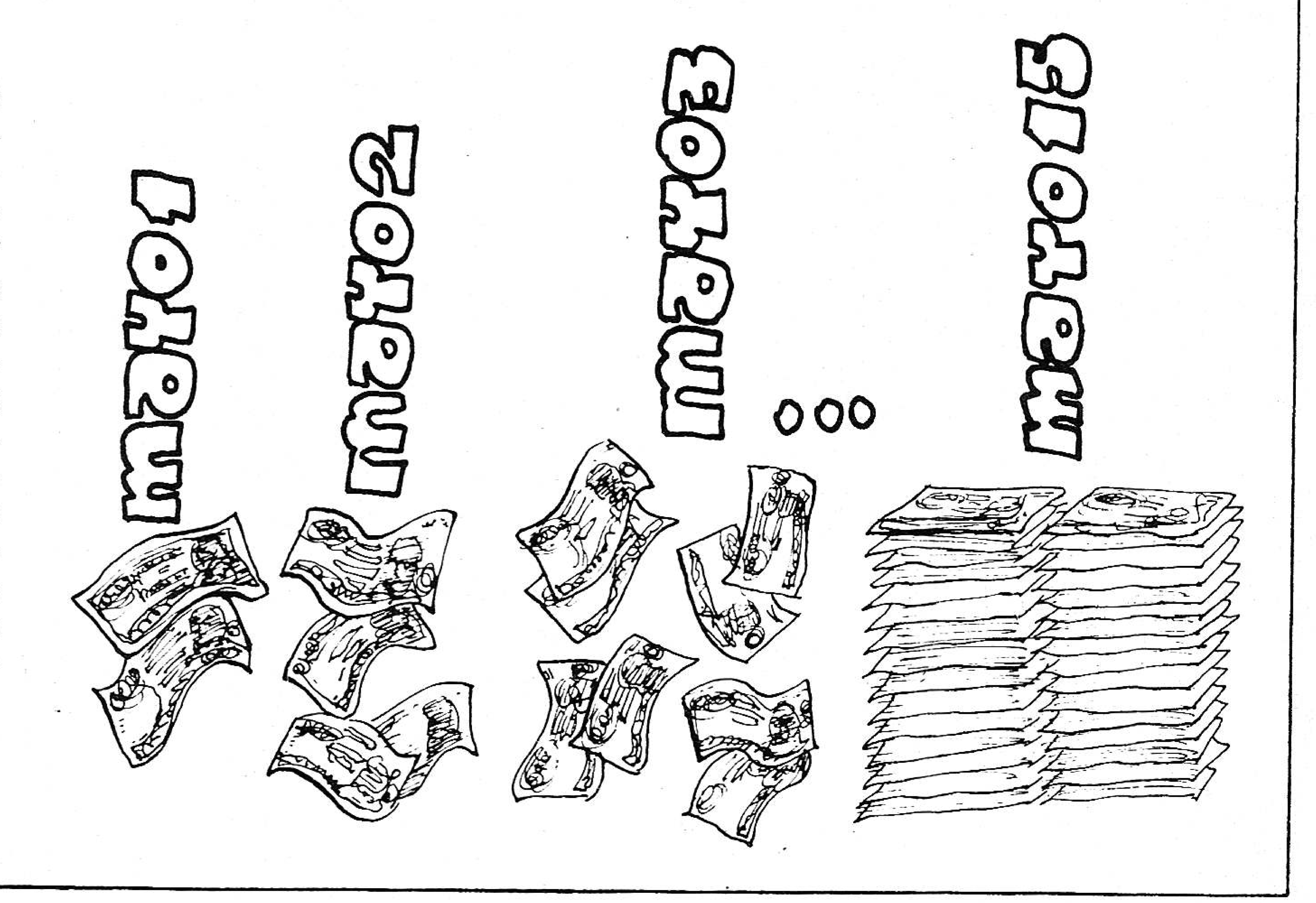
Nombre del código: /ARITH2/

Realicemos el siguiente programa, experimentemos con cambios en él:

```
100 PRINT "MEDIDA DE SOMBRERO EN FORMA DECIMAL"
110 PRINT 6+5/8;6+3/4;6+7/8;7;7+1/8;7+1/4;7+3/8
120 PRINT "MEDIDA DE PERFORACION"
130 PRINT 1/32,2/32,3/32,4/32,5/32,6/32,7/32,8/32
140 PRINT "DINERO DESPUES DE DUPLICARSE $1 EN 15 DIAS = $";2+15
150 END
```

Seguramente ahora usted está desalentado por la cantidad de tipeo que debe realizar para obtener una pequeña salida. El problema es que usted no puede escribir programas muy interesantes si las únicas palabras claves que conoce son PRINT y END. Por lo tanto intentaremos dos palabras claves extras (FOR y NEXT, las que discutiremos posteriormente) para ayudarlo a hacer de esta sesión en línea algo más interesante. **No esperamos que comprenda qué realizan estas palabras claves en este momento.** Sólo tipéelas como se indica:

NOTA: Los nombres código con teclas de barra oblicua (/) indican programas extras en línea.



Nombre del código: //MULTABLE//

```
10 PRINT "TABLAS DE MULTIPLICACION POR 10, 11, Y 12"
20 PRINT "------------------------------------------"
30 PRINT
40 FOR X=1 TO 12
50 PRINT X;"*10=";X*10,X;"*11=";X*11,X;"*12=";X*12
60 NEXT X
70 END
```

NOTA: PRINT sin nada luego de la misma produce lo que se denomina "alimentación de línea". Esto significa que el papel "alimenta" (pasa) una línea extra. Por lo tanto, el efecto de la línea 30 por sobre esto es colocar una línea en blanco en la SALIDA (OUTPUT) haciendo que resulte más nítido.

Realicemos una revisión

Aparecen de la siguiente manera diferentes formas de la sentencia PRINT:

```
20  PRINT 3+1; 4+3; "TIEMPO ATRAS"
36  PRINT "HOLA ALLI"
50  PRINT 900/450
123 PRINT 45
900 PRINT 10, 10*2, 10*3, 5[7*3, ((16+32)/8)*123
```

Si se utiliza más de una expresión (como en las líneas 900 y 20 antes mencionadas), se utilizan los siguientes símbolos de puntuación para separar la salida:

(,) Una coma separa la salida hasta 15 espacios

10 PRINT "2", "3", "4" da
2 3 4

10 PRINT 2, 3, 4 da (note el espacio para el signo)
2 3 4

(;) Un punto y coma imprime la salida toda junta:

10 PRINT "2"; "3"; "4" da
234

10 PRINT 2; 3; 4 da
2 3 4

•Una sentencia END es siempre necesaria en la última línea del programa. Consiste simplemente en un número de renglón y la palabra END.

•RUN es el comando que indica a la computadora que debe ejecutar todas las sentencias de su memoria. Como RUN no es una sentencia, no lleva un número de renglón.

•SCR significa **scratch** (borrar). Es un comando que borra el programa anterior de la memoria de la computadora. No lleva nunca un número de renglón.

•LIST es un comando que hace tipear a la computadora todas las sentencias que guarda en su memoria en este momento. No lleva nunca un número de renglón.

2-3 Sentencias que utilizan la palabra clave LET

Es tiempo de elección y recién se han contabilizado los votos de los tres candidatos principales. Flamboyant tiene 8497 votos, Handsome tiene 7231 votos y Moderate encabeza el grupo con 9821 votos. En la figura observamos como los trabajadores en los cuarteles centrales dispuestos para la elección han "almacenado" esta información en el pizarrón.

Nuestra figura ilustra tres espacios o **ubicaciones** en el pizarrón, denominados F, H y M. Podemos pensar en F, H y M como

identificaciones o **rótulos** en el pizarrón. Al lado de cada una de estas identificaciones se escribe el número de votos "almacenados" en nuestra memoria del pizarrón. Por supuesto, estos números pueden borrarse en cualquier momento, y se pueden colocar nuevos en cada ubicación. Ahora utilicemos esta figura para lograr sentir lo que sucede en las memorias de una computadora. Nosotros también podemos "almacenar" números en la memoria de una computadora. Para conocer donde se guardan estos números también debemos utilizar **identificaciones** para las diferentes ubicaciones de la memoria.

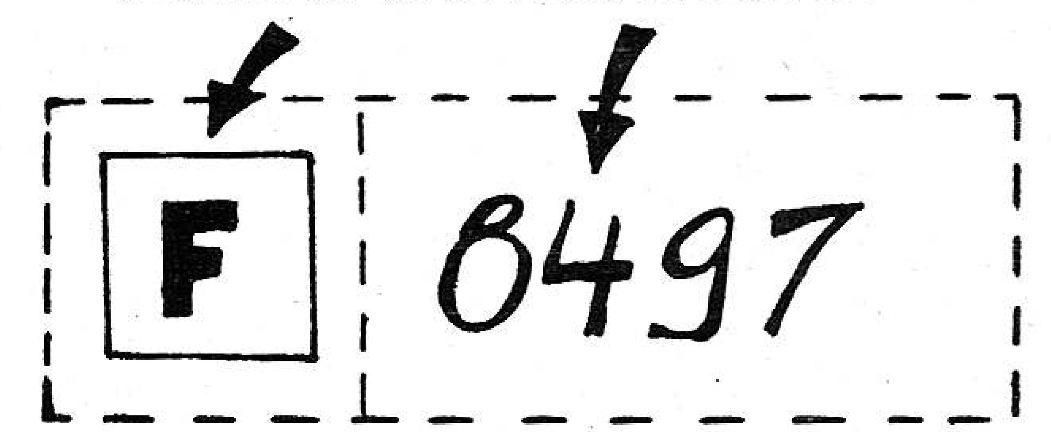
La sentencia LET en BASIC realiza ambas cosas a la vez.

- •Suministra una identificación a la ubicación de la memoria.
- •Almacena un número en esta ubicación de la memoria.

Por ejemplo, la sentencia:

```
20 LET F = 8497
```

IDENTIFICACION CONTENIDOS



- •Suministra la identificación F a una ubicación en la memoria de la computadora.
- •Almacena el número 8497 en la ubicación de la memoria que tiene esa identificación. El número 8497 es denominado los **contenidos** de la ubicación F de la memoria.

Algunas veces las identificaciones son comparadas a los nombres colocados en los buzones tal como se ilustra en la figura. Nótese que la **identificación** es muy diferente a los **contenidos** del buzón.

Un buzón tiene la **identificación** Smith, pero contiene una carta.

Nosotros podríamos llamar a la identificación Smith una **variable** por cuanto el material colocado en el buzón "Smith" puede **variar:** un día una carta, el día siguiente una revista.

En forma similar, las identificaciones utilizadas para las ubicaciones de la memoria en una computadora se denominan **variables.** La razón de esto es que se pueden almacenar números diferentes en una ubicación de la memoria de una computadora, y por lo tanto su contenido puede **variar.** En BASIC los nombres que utilizamos



para las identificaciones normalmente son letras tales como A, B, C, X, Y.

Las memorias reales de la computadora no se asemejan a pizarrones o buzones, y eso dejémoslo bien sentado. Sin embargo, una persona que desee programar una computadora no tiene que conocer sobre la construcción real de las memorias, y para nuestro propósito la figura del pizarrón es mejor.

F ~~230~~ ~~565~~ 8497

H ~~114~~ ~~429~~ 7231

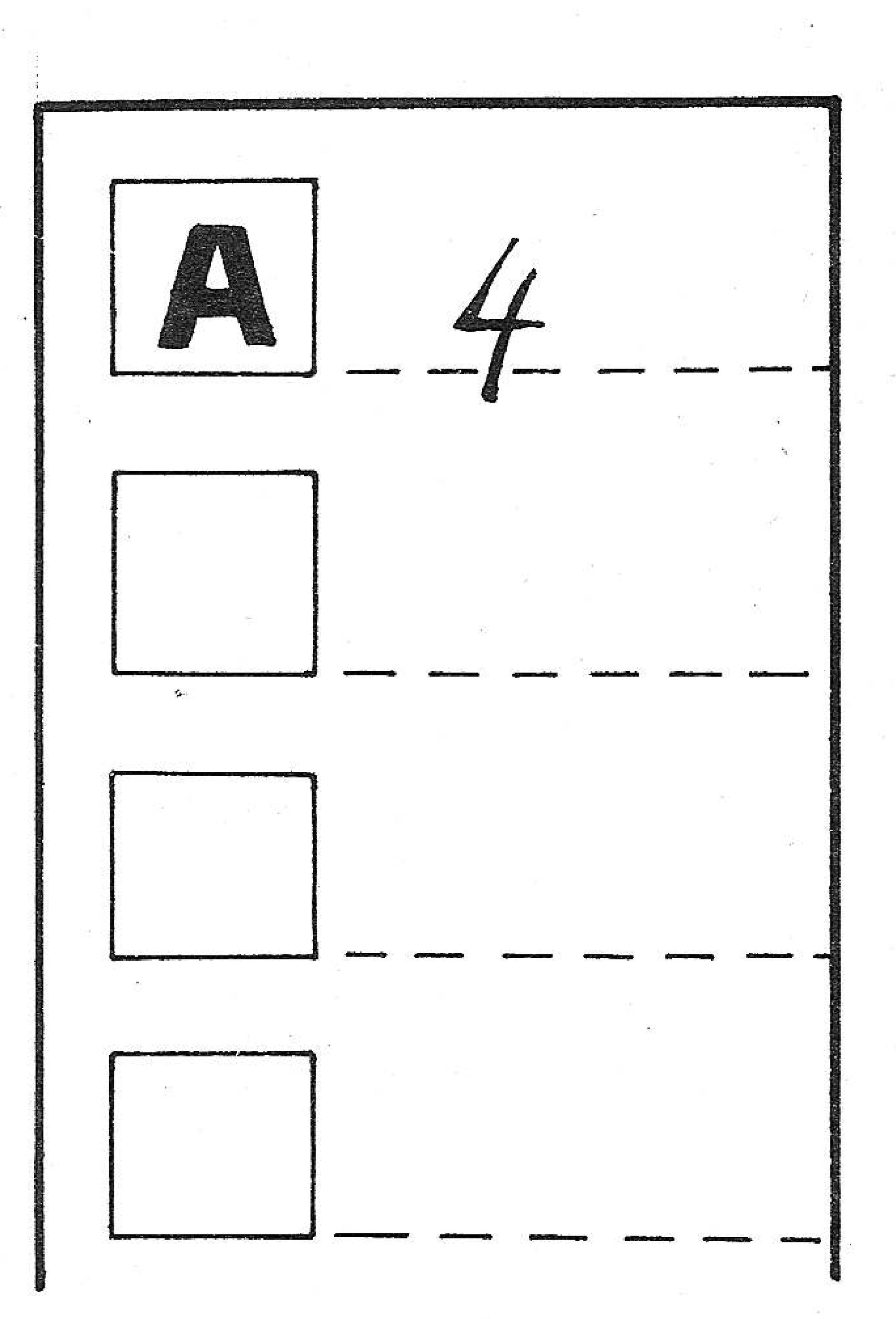
M ~~375~~ ~~745~~ 9821

Observamos que nosotros podemos **borrar** el número que se encuentra al lado de la identificación y colocar un número nuevo. Esto es exactamente lo que hacen las computadoras en sus memorias electrónicas. Si colocamos un número nuevo en la misma ubicación en la que estaba el número anterior, el primer número se borra.

Si un programa BASIC dice:

```
10 LET A = 4
```

podemos imaginar que la memoria de la computadora se asemeja a lo siguiente:

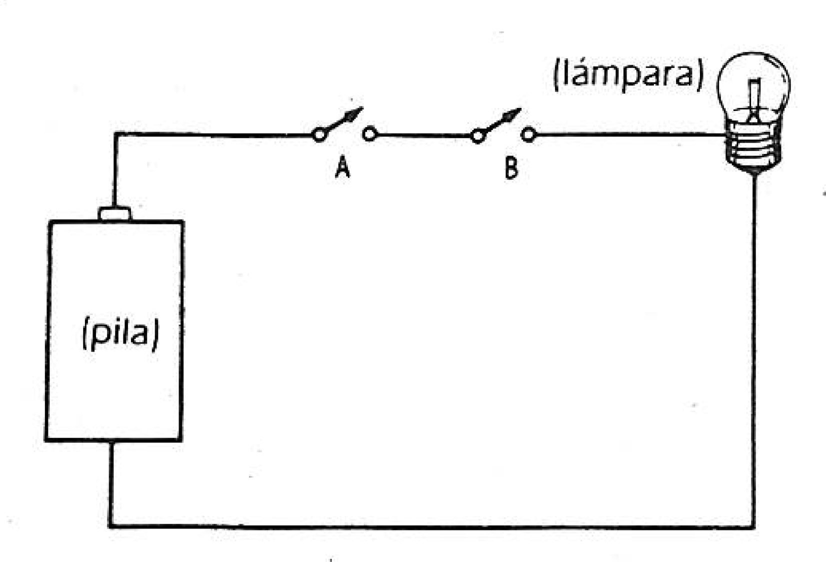


Curso de computadoras digitales

Capítulo 6:

El circuito "Y"

Podemos realizar fácilmente un circuito Y con un par de interruptores conectados en serie como el que se indica a continuación:



La única manera de encender la lámpara es cerrar ambos interruptores A y B. Ninguna otra combinación de encendido y apagado permitirá a la corriente llegar a la lámpara. Todas las posibles combinaciones de interrupción (conmutación) se pueden resumir en una tabla como la siguiente:

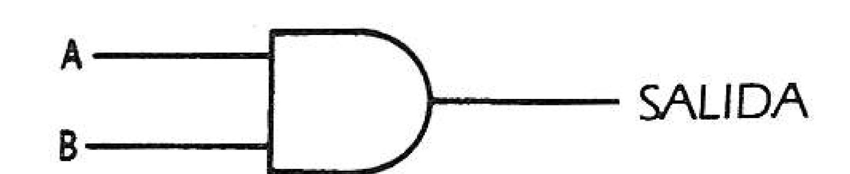
Interruptor		**Lámpara**
A	B	
no	no	no
no	si	no
si	no	no
si	si	si

Si asignamos el bit binario 0 al estado no y el bit 1 al estado sí, y llamamos a la lámpara salida, la tabla se transforma en:

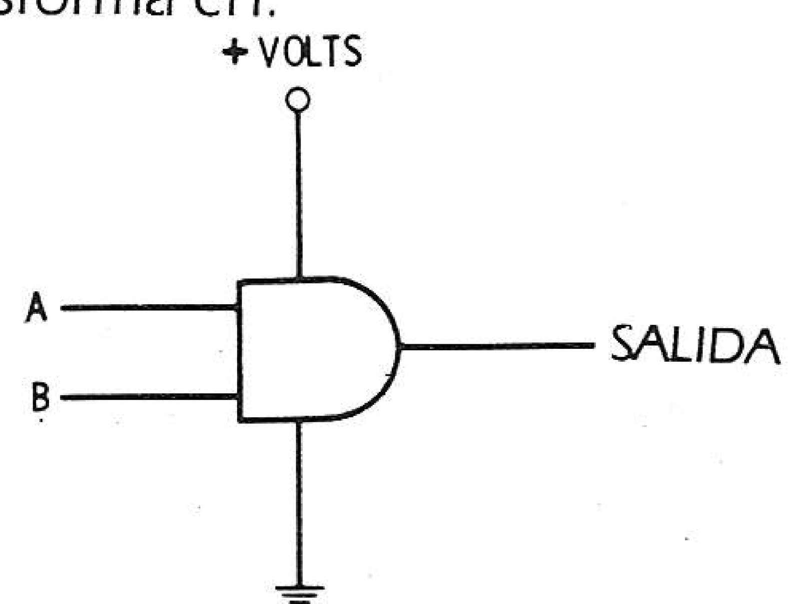
A	**B**	**SALIDA**
0	0	0
0	1	0
1	0	0
1	1	1

Esto, por supuesto, es idéntico a la tabla para la función Y a la que echamos un vistazo anteriormente.

El símbolo lógico para el circuito Y tiene la siguiente forma:

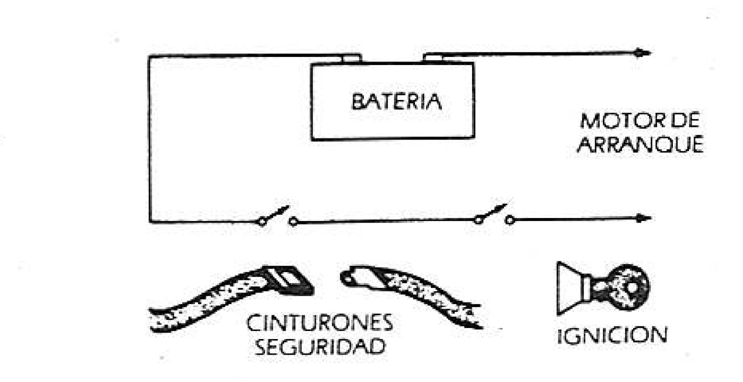


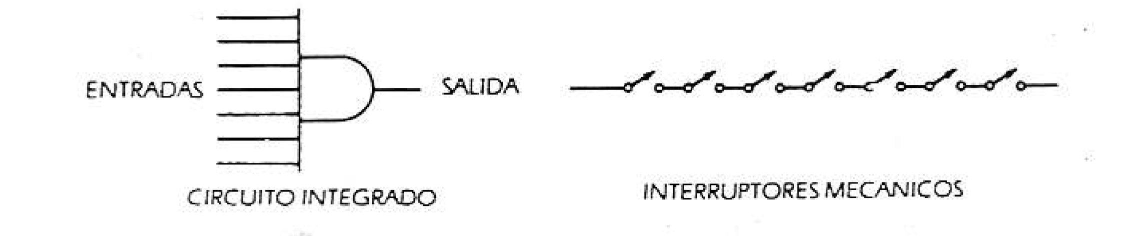
Los circuitos lógicos electrónicos del mundo real requieren energía eléctrica, y cuando se agregan estas conexiones, la puerta Y se transforma en:



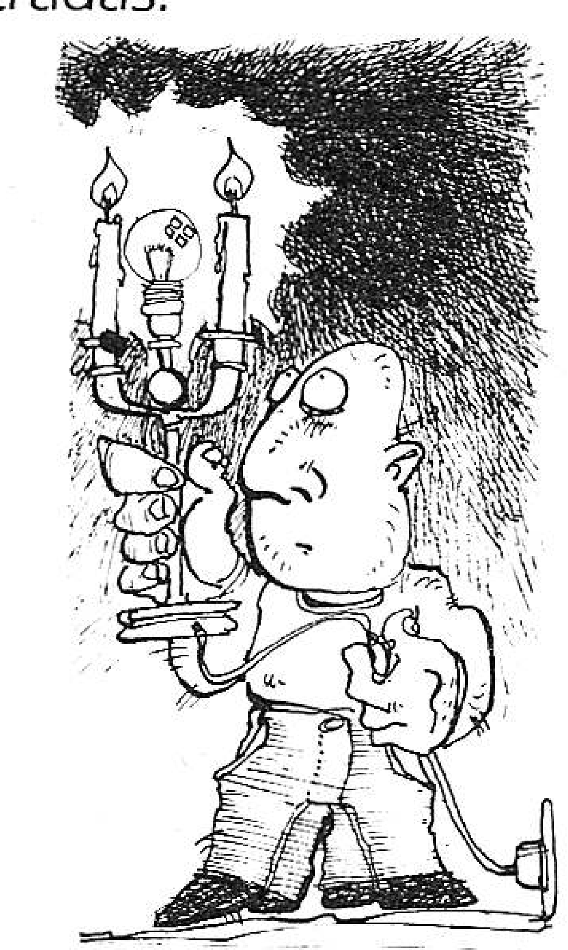
En la práctica, las conexiones eléctricas raramente, si no nunca, se indican en los diagramas de los circuitos lógicos dado que tienden a confundir las cosas. Además, resulta fácil recordar que hay que conectar la corriente eléctrica a cada chip de lógica. Las conexiones **entre** chips es lo que a usted le debe importar.

La característica más importante del circuito Y es que es un elemento verdadero de toma de decisiones. Esto hace posible un sinnúmero de aplicaciones simples tales como el monitoreo de los cinturones de seguridad del automóvil.





Y la puerta Y no está solamente limitada a las dos entradas, sino que las versiones de los circuitos integrados frecuentemente tienen tres, cuatro o aún más entradas. El sistema de monitoreo de los cinturones de seguridad para un avión formaría una puerta Y con docenas de entradas.



Punto de verificación

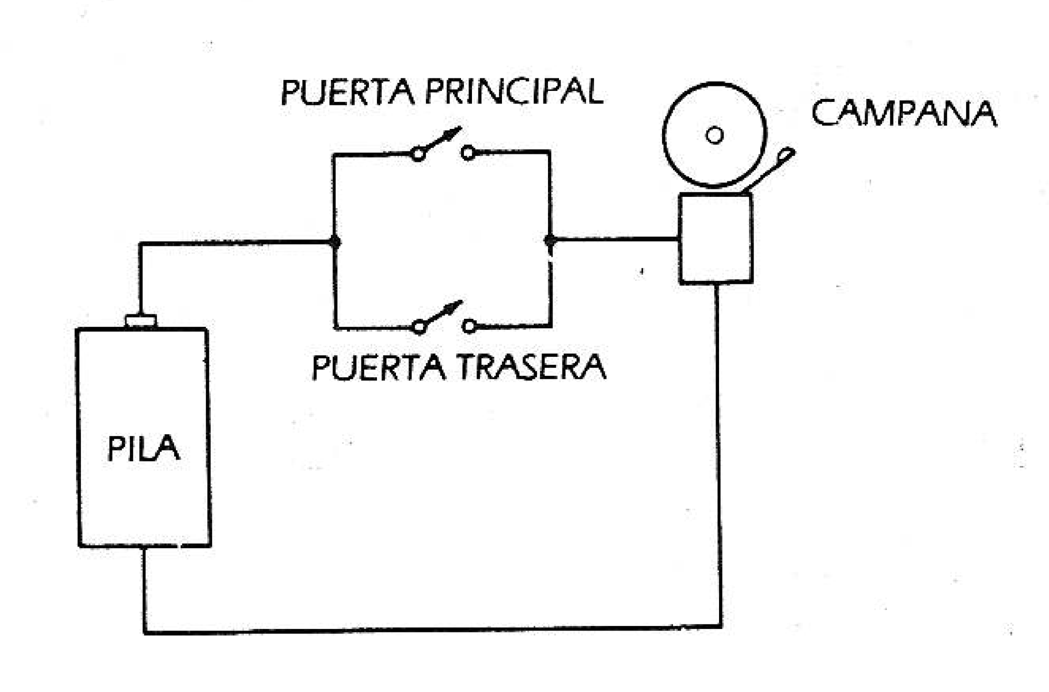
1. Una puerta Y tiene por menos .... entradas.
2. Si ambas entradas de una puerta Y son lógica 1, la salida será lógica ....
3. ¿Puede una puerta Y tener más de diez entradas?

Respuestas

1. Dos
2. 1
3. Por supuesto.

El circuito "O"

Los interruptores de una puerta Y indicados anteriormente pueden ser fácilmente re-adaptados desde una serie a una conexión en paralelo para producir un circuito simple O. Se vería de la siguiente forma:



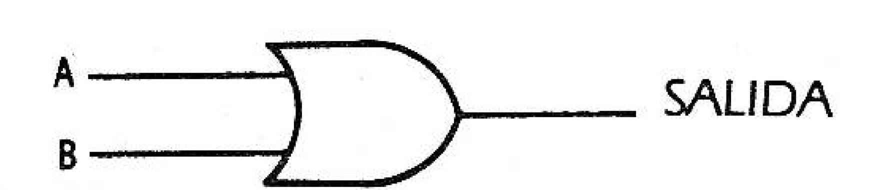
El circuito simple encenderá la lámpara cuando cualquier interruptor A o B o ambos se cierren. Las diferentes combinaciones de los interruptores pueden resumirse de la siguiente manera:

Interruptor		**Lámpara**
A	B	
no	no	no
no	sí	sí
sí	no	sí
sí	sí	si

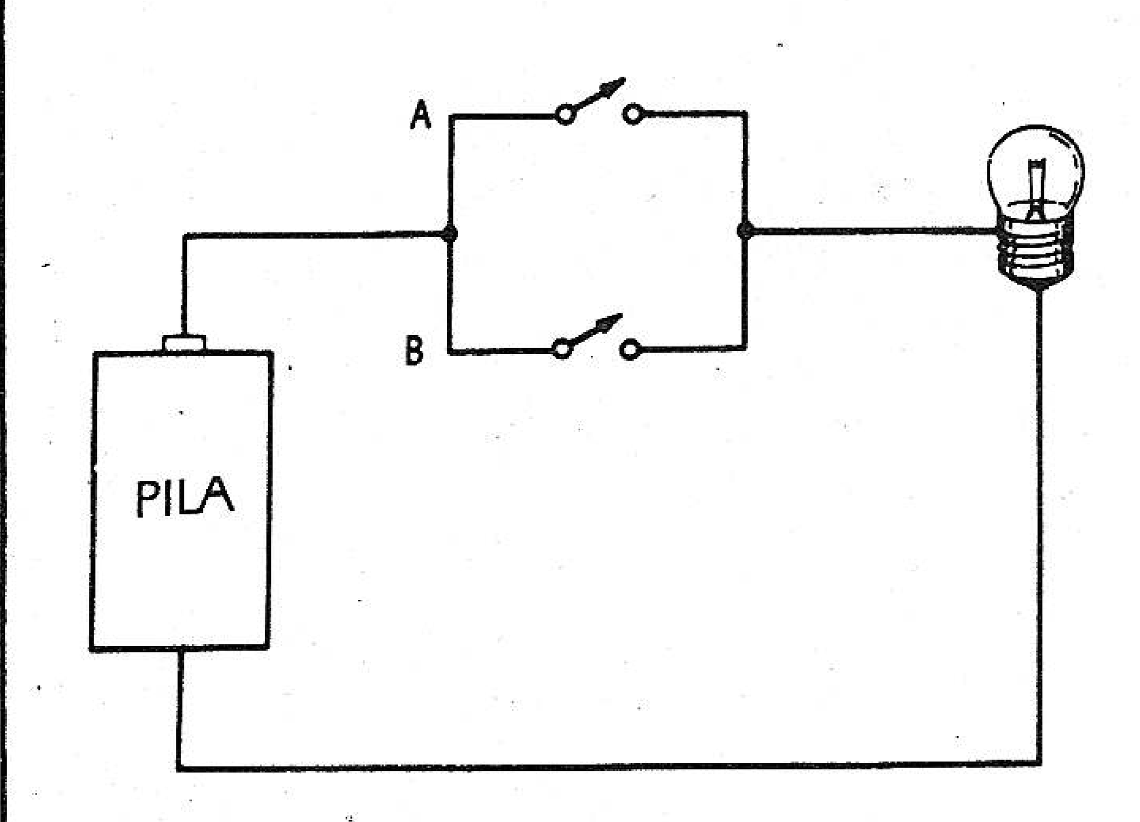
Esta tabla se puede simplificar a la versión binaria más familiar que definimos anteriormente:

A	**B**	**SALIDA**
0	0	0
0	1	1
1	0	1
1	1	1

El símbolo para la puerta O se ve de la siguiente manera:



Al igual que el circuito Y, la puerta O es un circuito lógico real de toma de decisiones. En la práctica puede ser tan simple como un par de pulsadores separados que controlan una campana:



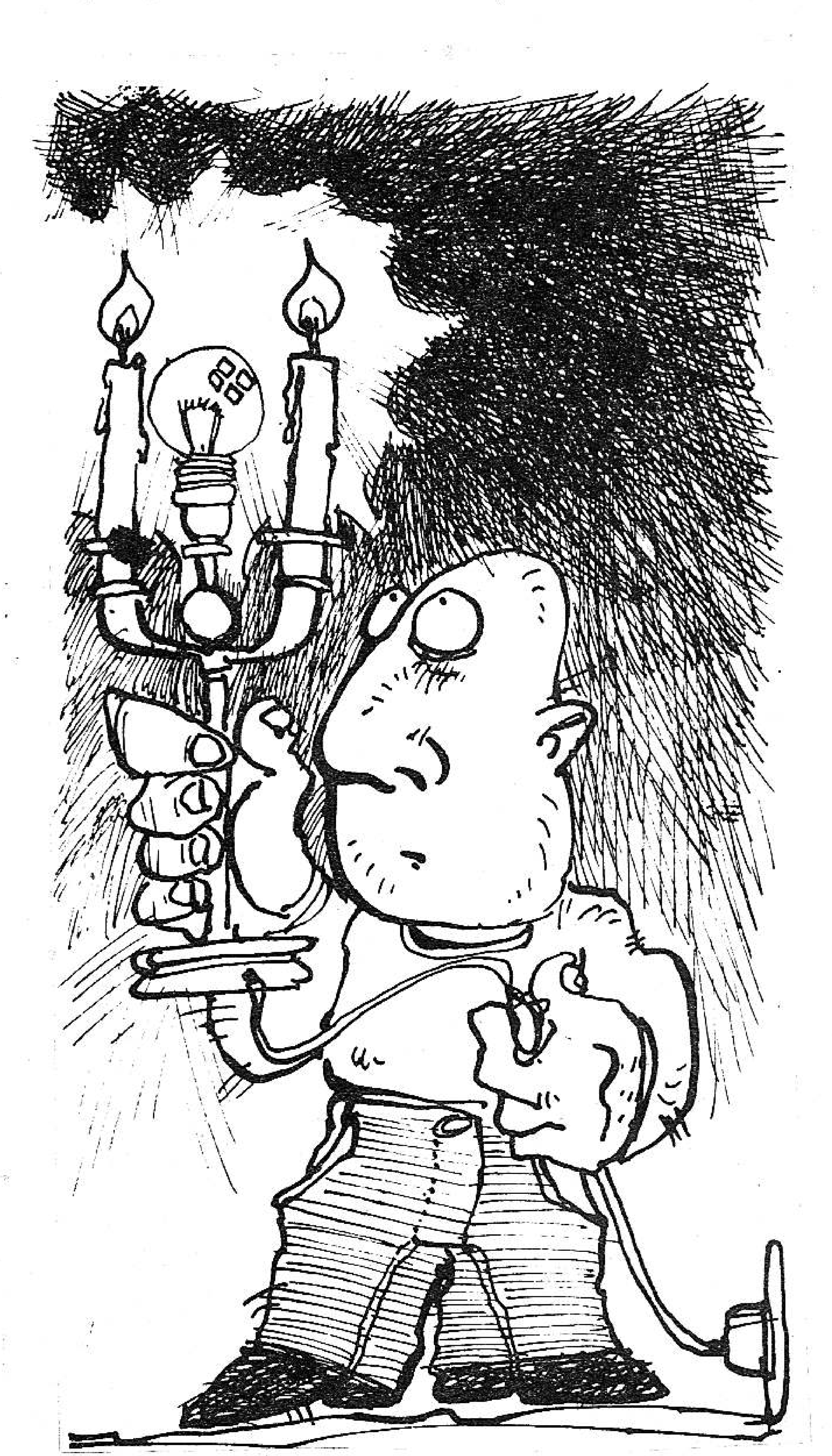
¿Puede tener una puerta O más de dos entradas? Por supuesto, las versiones de los circuitos integrados comúnmente tienen varias entradas y los circuitos de conmutación pueden tener docenos (vea el dibujo que se ilustra a continuación).

Punto de verificación

1. Una puerta O requiere por los menos .... entradas.
2. Si una entrada de una puerta O es una lógica 1, entonces la salida será lógica ....
3. ¿Puede tener una puerta O tres entradas?

Respuestas

1. Dos.
2. 1.
3. Sí.

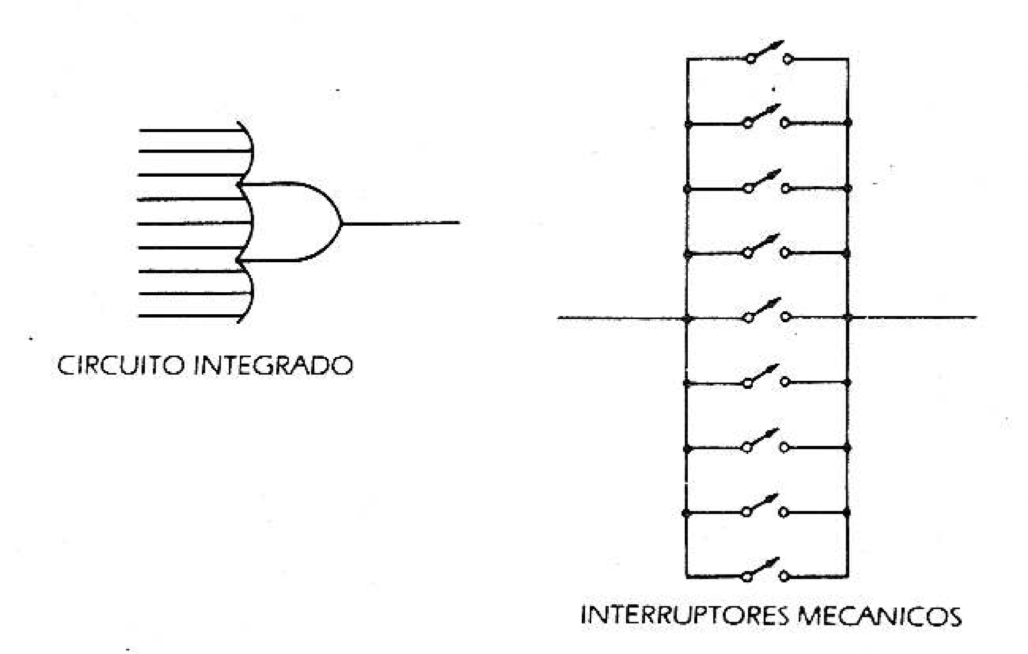


El circuito "NO"

El circuito NO tiene una sola conexión de entrada y, como usted ya sabe, el estado de la salida es siempre opuesto al estado de la entrada. Por lo tanto una entrada 0 suministra una salida 1 y viceversa. Aquí nuevamente tenemos una tabla de la realidad de los expresado:

A	**SALIDA**
0	1
1	0

Como usted puede ver, el circuito NO cumple con el papel lógico de complementación o inversión. Normalmente, en realidad, los circuitos NO son denominados inversores. El símbolo de un circuito NO es una triángulo conectado a un pequeño círculo.

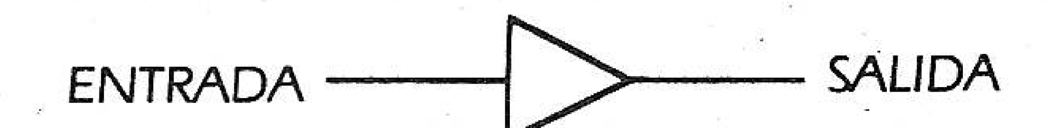


A propósito, cada vez que usted vea un pequeño círculo en la salida de un símbolo lógico, esto significa que la salida normal de ese símbolo está invertida.

Si usted aún no está convencido de que el circuito NO es práctico, sólo recuerde el teorema de Morgan. Por cuanto un inversor puede convertir lógica positiva a negativa (o viceversa), podemos realizar puertas O de puertas Y!! Es imposible enfatizar lo enfatizado de cuán importante es esta capacidad. En principio, significa que usted puede, en realidad, construir una computadora con nada más que puertas Y e inversores!! Más tarde.

El circuito "SI"

El circuito SI es tan simple que frecuentemente no se lo describe como un circuito lógico en los libros que desarrollan temas de computación. Pero en realidad es más que un circuito lógico, y su diagrama es el que se indica a continuación:

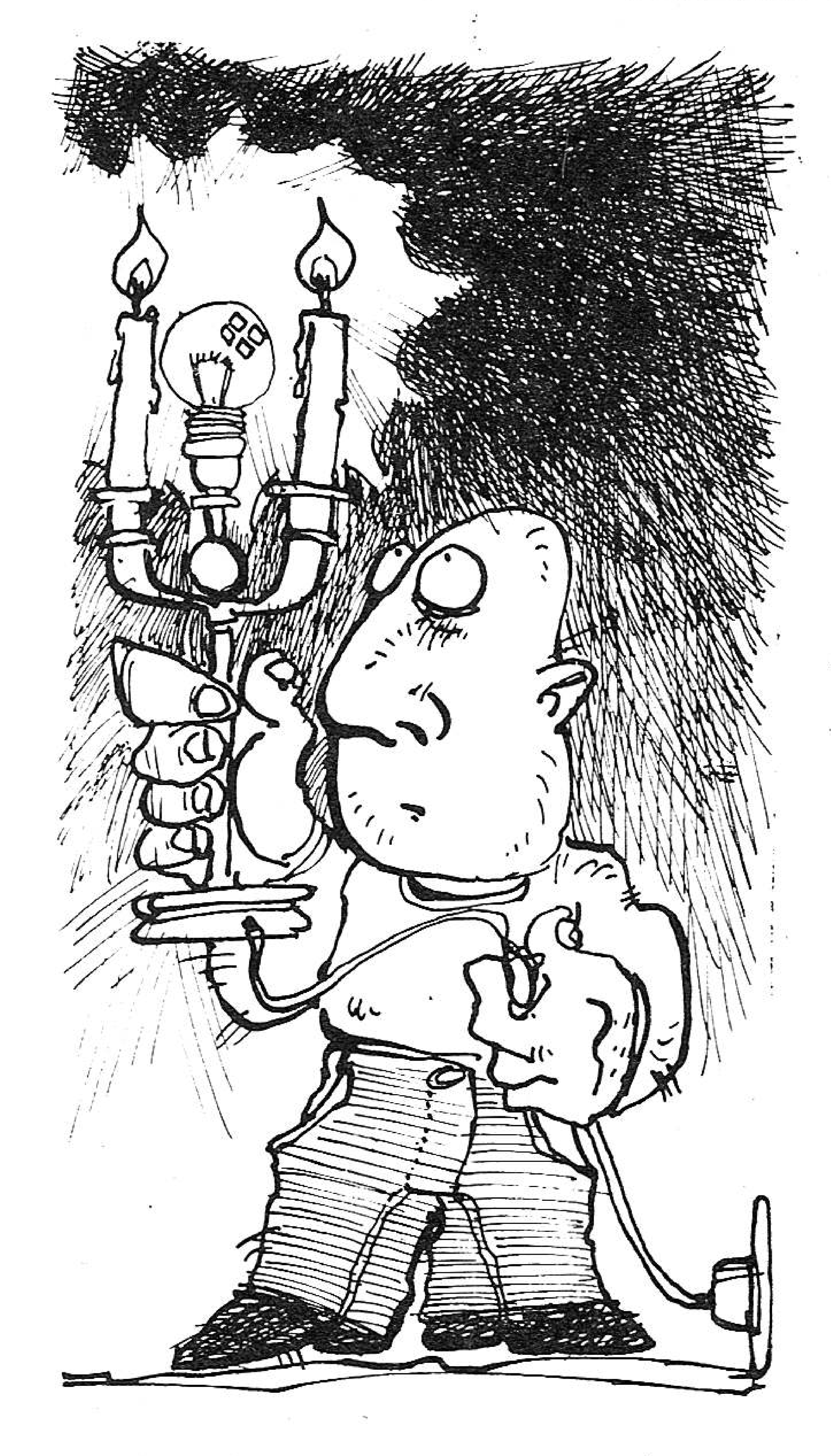


Como su nombre así lo indica, la salida del circuito SI es siempre la misma que la entrada. Por lo tanto, si 1 es la entrada, 1 será la salida. Aquí tenemos la tabla que nos indica esta situación:

A	SALIDA
0	0
1	1

Actualmente usted probablemente pensará que el circuito SI es casi tan lógico como un eco!! Es cierto, un circuito SI puede ser en principio tan electrónico como un simple cable de cobre. Pero en la práctica los circuito SI son frecuentemente amplificadores que se "quejan" de las señales muy débiles, o lo que es más importante, aislan un circuito lógico de otro.

Los circuitos SI que fundamentalmente aislan se los denomina **separadores**. Un ejemplo de separador es un transistor conectado a la salida de un circuito lógico de baja potencia para permitirle encender una lámpara. El circuito no



puede energizar la lámpara por sí mismo, dado que la corriente que fluye a través de la lámpara (y por lo tanto a través del circuito) eventualmente sobrecalentaría y posiblemente dañaría o aún destrozaría el circuito delicado. Un transistor adecuado, sin embargo, podría "embeber" el calor en exceso y accionar la lámpara cuando es encendida por el circuito lógico.

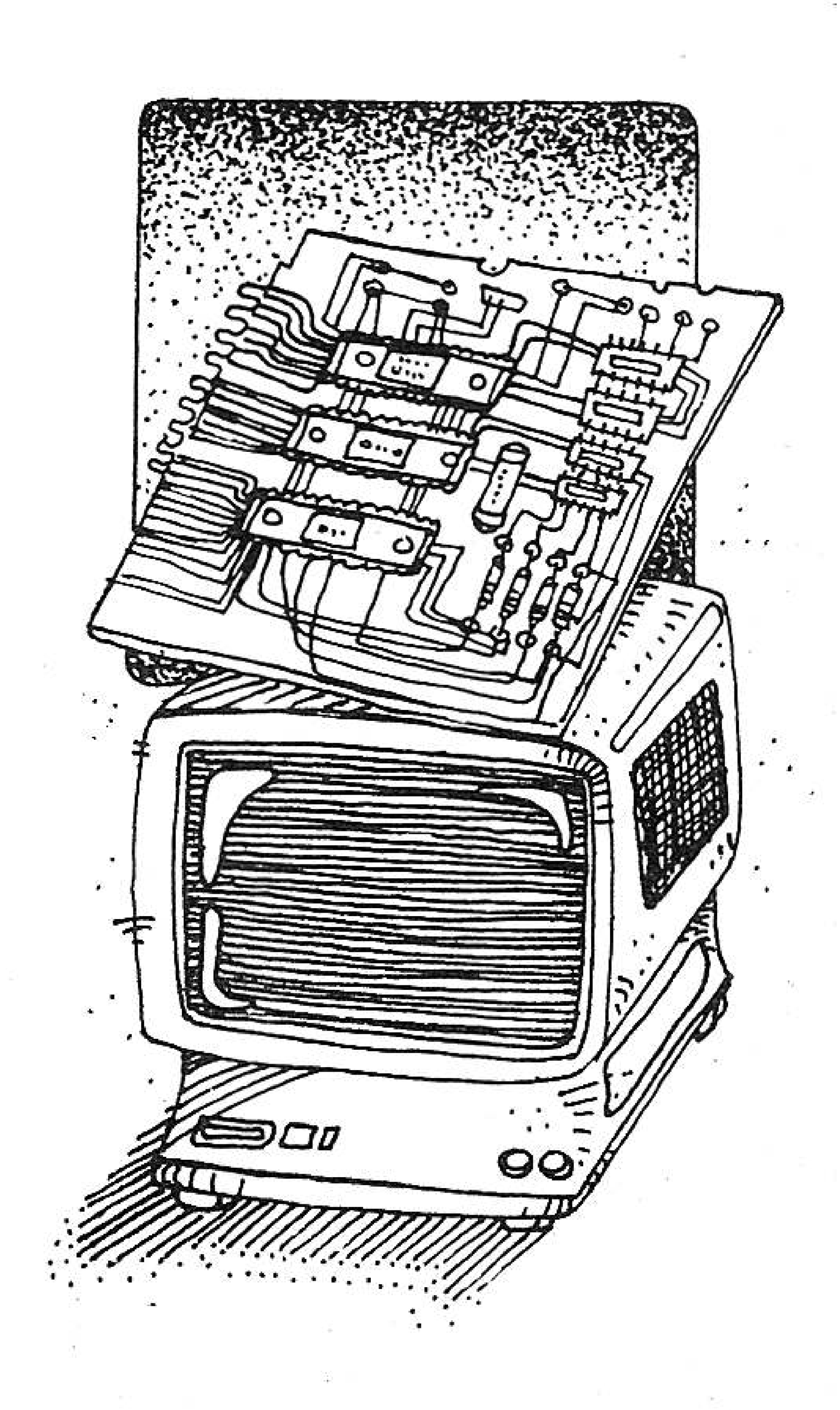
Punto de verificación

1. ¿Es realmente un elemento lógico un circuito SI?
2. Escriba la tabla para el circuito SI.
3. Otros nombres más comunes para el circuito SI son: .... y ....

Respuestas

1. Sí.
2. Verifique su tabla con la suministrada anteriormente.
3. Separador y aislador.

Curso de electrónica digital



¿Cómo representa una calculadora a todos los sistemas digitales?

Este es el fin de nuestra visión general inicial de los sistemas digitales en la forma de una calculadora prototipo. Antes de pasar a nuestro siguiente capítulo, hagamos una pausa y pensemos cómo las ideas que hemos visto se pueden generalizar de una calculadora prototipo a los sistemas digitales.

La generalidad más importante, tal como se indicó anteiormente, es que todos los sistemas electrónicos digitales modernos operan como una calculadora reduciendo la información y las tareas a **términos** muy **simples** – a una cuestión de conectar o desconectar (on/off), si o no, uno o cero (1 ó 0).

Manejar la información y las tareas de cualquier complejidad requiere emplear **grandes cantidades** de tales sentencias y tareas simples, realizarlo rápidamente utilizando **esquemas de código** por los cuales muchas partes simples de información pueden representar una cantidad de información más compleja. Usted verá este patrón en cada sistema electrónico digital.

¿Cómo se adecúa la electricidad a los requerimientos del sistema digital?

Aún no lo hemos considerado, pero el hecho es que usted puede construir un sistema digital perfectamente funcional **sin utilizar electricidad en absoluto**. No existe nada en nuestra definición de los sistemas digitales que hablen de electricidad – solamente separar la información en partes, utilizar dígitos numéricos, etc. Un ejemplo de lo que hemos citado de un sistema que no utiliza electricidad —aún más, un sistema no binario— es nuestra calculadora mecánica.

Otro ejemplo, uno más actualizado, es el de los sistemas digitales binarios que emplean dispositivos que conmutan **líquidos o gases** que fluyen en pequeños tubos. Los llamamos sistemas "fluídicos".

Pero la razón por la cual se ha empleado **la electricidad** para los sistemas digitales con tanto éxito es que **los circuitos de conmutación eléctrica** —que son relativamente simples y de bajo costo comparados con otros circuitos eléctricos— se pueden utilizar para manejar la información así como las tareas más simples involucradas en los sistemas digitales binarios. Estos circuitos son los más rápidos así como el método más conveniente que conozcamos para tales propósitos.

¿Por qué los circuitos integrados se adecúan tan bien?

Los primeros sistemas eléctricos digitales utilizaban relés electromecánicos que en realidad contenían pequeños interruptores mecánicos del tipo de los que nos hemos estado imaginando en los circuitos de conmutación. Posteriormente, los sistemas eléctricos digitales utilizaron tubos de vacío en cambio de lo mencionado anteriormente. Muy pronto el transistor apareció como el reemplazo y

más tarde los circuitos integrados de semiconductores.

Y aquí nuevamente parece que somos testigos de un casamiento realizado en el cielo. Tal como veremos más claramente, (posteriormente en esta evaluación), **los circuitos integrados se adaptan naturalmente** para reducir circuitos de conmutación simples a tamaños microscópicamente pequeños y colocar una cantidad incontable de miles de ellos en un espacio increíblemente pequeño, disminuyendo significativamente el costo por circuito.

Esta capacidad lleva a los circuitos integrados directamente a los brazos de los sistemas digitales – los que, como hemos visto, involucran **muchas** tareas simples y partes de información. La electrónica del semiconductor integrado es la mejor forma que hemos encontrado para implementar los sistemas digitales – y mejora en la medida en que constantemente la tecnología del circuito integrado evoluciona de manera tal que se pueda colocar más y más circuitos en un pedazo de material de sílice.

¿Qué es lo que hacen todos los sistemas?

De aquí en más, pasaremos a una generalización extraída de nuestro modelo de calculadora, la que se ilustra en la figura 6.1. Esta generalización se compone de dos ideas. Primero, las únicas cosas que realiza cualquier sistema, o que puede realizar, es manipulear información y hacer tareas (o ambos). Es decir que todo lo que se realiza en cualquier sistema es el manejo de diferentes formas de información, quizás asociado con la realización de trabajo.

¿Cómo se organizan todos los sistemas?

Y segundo, todos los sistemas se organizan de la misma manera. Realizan sus tareas en los mismos pasos o etapas generales. Primero, **captan**, (o detectan, o aceptan) información en diferentes formas del mundo exterior, y la convierten a formas de información que puedan ser manejadas en el sistema. Luego toman **decisiones** basadas en la información de entrada, entendiendo por esto último que procesan o manejan la información. Al hacerlo, pueden **almacenar** o recordar alguna parte de la información por algún tiempo, o procesarla a consecuencia de o como resultado de otra información que se encuentra **almacenada** permanentemente. Y finalmente, toman la información resultante y **actúan** con ella en el mundo exterior convirtiéndola nuevamente en formas de información externa, y quizás ejerciendo alguna forma controlada de trabajo o energía. Piense en cualquier sistema que le agrade y este concepto de organización universal puede construirse para su aplicación.

Por ejemplo, nuestro teclado de la calculadora y el codificador **captan** información y la convierten a una forma interna. Varios subsistemas **deciden** y **almacenan**. Y la parte del sistema correspondiente al decodificador y al visor (display) convierten la información interna resultante en la **acción** deseada de mostrarle los números en el visor. Este sistema "electrónico digital" está, por supuesto, manejando la información en forma digital.

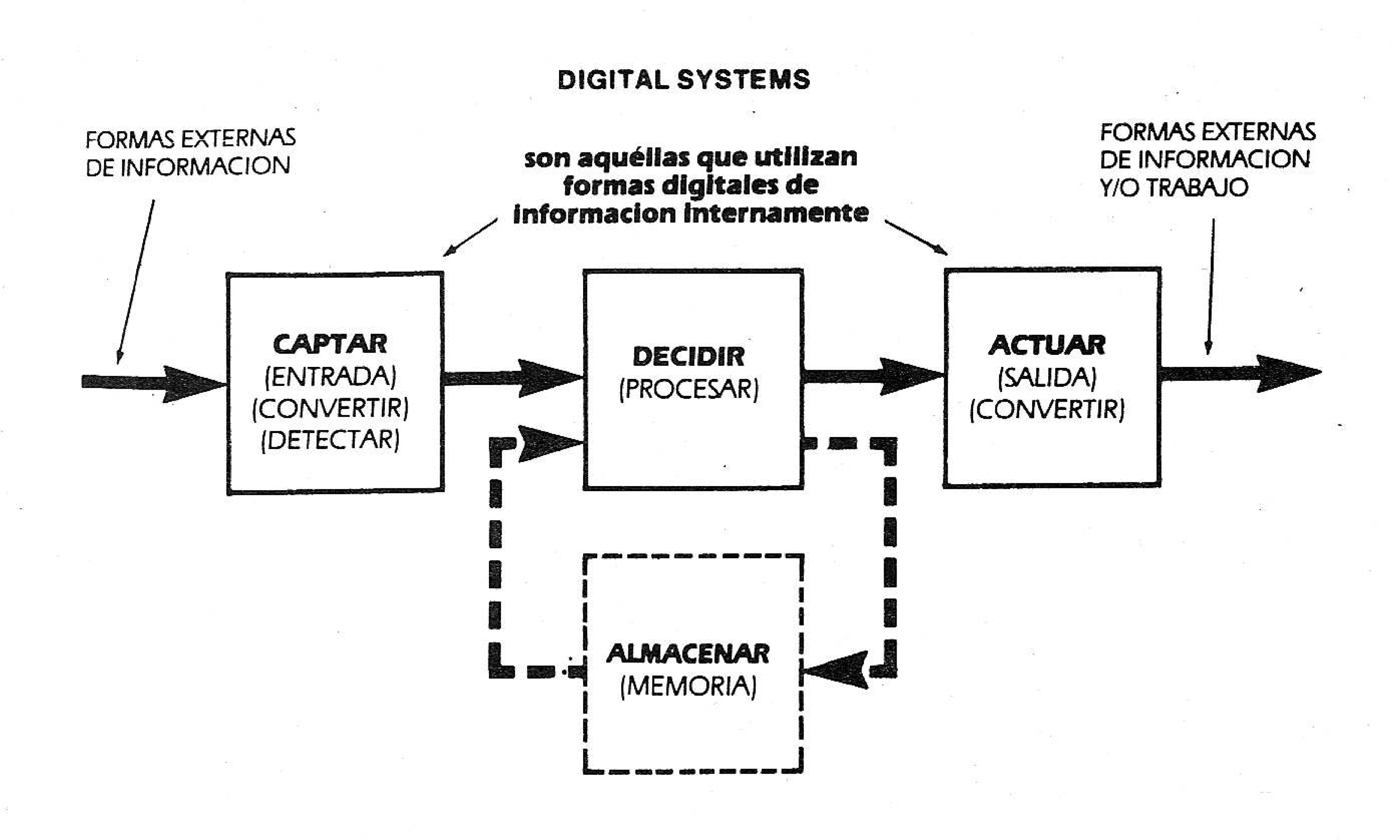


Figura 6.1 La organización universal del sistema. Todos los sistemas manipulan información y/o realizan trabajos utilizando las mismas tres o cuatro etapas.

¿Cómo distinguir los sistemas digitales de los otros?

El significado de este concepto universal del sistema es que nos muestra que los **sistemas digitales** son aquéllos que manejan **la información en la forma digital**, lo que hemos visto significa en la forma de dígitos, pequeñas piezas separadas de información. Hay sólo otro método general para manejar la información, y se lo llama "analógico". Posteriormente en nuestra evaluación, estudiaremos las diferencias entre estos dos tipos de información y los dos tipos de sistemas que resulten de ella.

Y hemos recorrido un largo camino! Hemos pasado de la comprensión general de una calculadora de mano, a través de una introducción al concepto de los sistemas digitales, al entendimiento de los conceptos unificados de todos los sistemas. Esto suministrará una base de comprensión a medida que procedamos a profundizar en los sistemas digitales y veamos **cómo** realizan las cosas que hemos estado tratando.

Tómese un respiro

A medida que vayamos integrando los diferentes conocimientos, será una buena idea para usted detenerse y tomarse un respiro. Y antes de seguir revisando los temas que se desarrollarán en cada una de nuestras ediciones, repase y estudie cada una de las partes que no le quedaron demasiado claras. Esto se debe a que todas estas ideas son necesarias para la comprensión de lo que veremos más adelante. Siempre habrá un glosario y una revisión para ayudarlo a clarificar y fijar sus conocimientos.

Revisemos los temas tratados hasta aquí

1. ¿Cómo sabe el circuito de la calculadora qué tecla causa una señal en la línea de entrada del teclado?
a) Para cada tecla hay una línea de entrada diferente.
b) Para cada tecla hay una línea de rastreo diferente.
c) Verificando qué línea de rastreo está activada/conectada cuando se recibe la señal.
d) ByC anteriormente mencionados.

2. ¿Por qué titilan los numerales en el visor (aunque más rápido de lo que usted puede ver)?
a) Usan corriente alterna.
b) Están apagados mientras que el controlador re-verifica las entradas para verificar que se recibió una señal.
c) Las salidas de segmento pueden transmitir sólo un numeral por vez.
d) El registro del visor sólo almacena un dígito por vez.

3. ¿Cómo es capaz el controlador de realizar tantas cosas diferentes en momentos diferentes?
a) Contiene un circuito especial y diferente para cada tarea que debe controlar.
b) Realmente no controla los otros subsistemas — ellos funcionan independiente y automáticamente.
c) Sólo repite el mismo proceso para cada tarea que debe realizar.
d) Se le dice qué debe hacer mediante instrucciones leídas de la memoria del microprograma.

4. ¿Cómo se mantienen coordinadas las operaciones en todos los subsistemas?
a) Cada subsistema tiene una pequeña unidad "reloj" (temporal).
b) Mediante señales de control del controlador.
c) Mediante señales en las líneas de rastreo.
d) Por pulsos de tiempo en tres configuraciones llamadas fases.

5. Cuando se pulsa la tecla de "resultado", cómo sabe el controlador qué operación aritmética debe realizar?
a) Verifica una nota que realizó respecto a esto en el registro de finalización.
b) La instrucción corriente del microprograma contiene esta información.
c) Hay un lugar en cada registro numérico para el signo menos, más, multiplicación, etc.
d) Ya ha realizado la necesaria operación y está esperando exhibir el resultado.

6. Todas las operaciones aritméticas en la calculadora se manejan por una unidad que sólo puede:
a) Sumar
b) Restar
c) Comparar dos números.
d) A y B mencionado anteriormente.

7. El circuito de conmutación que controla un cable en un sistema digital binario:
a) Está tanto "conectado" (ON) o "desconectado" (OFF).
b) Frecuentemente está "parcialmente" conectado o desconectado.
c) Puede controlarse por otros circuitos de conmutación.
d) A y C antes mencionado.

8. Un circuito de conmutación binario puede indicar una elección entre cuántas alternativas (en un cable, en un momento)?
a) Una
b) Dos
c) Diez
d) Depende del diseño del circuito.

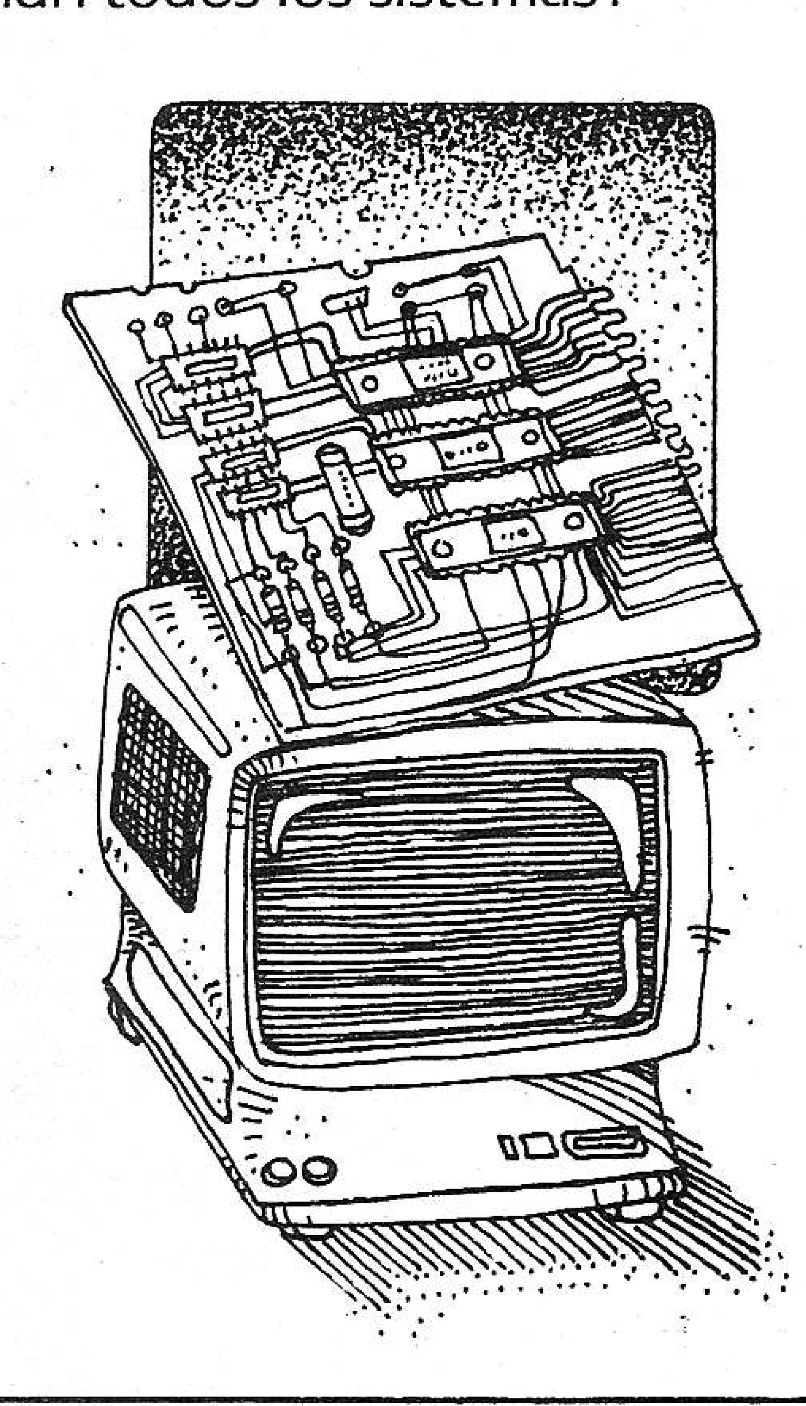
9. ¿Qué es un "bit" en un sistema digital?
a) Un dígito binario (1 ó 0).
b) La unidad básica de información.
c) La parte más pequeña de información.
d) Todo lo anteriormente mencionado.

10. ¿De dónde surge el nombre de electrónica "digital"?
a) Usted ingresa mediante teclas números con sus dedos (dígitos).
b) Todos los sistemas digitales utilizan dígitos binarios (bits).
c) Todos los sistemas digitales utilizan algún tipo de dígitos numéricos. (decimal, binario, etc.)
d) Todos los sistemas digitales tienen visores de números digitales como en las calculadoras.

11. ¿Qué número binario representa "siete"?
a) 1111111
b) 7
c) 07
d) 0111

12. ¿Cuál de las siguientes opciones es la que maneja información y posiblemente realiza trabajos o tareas?
a) Todos los sistemas.
b) Solamente los sistemas digitales.
c) Solamente los sistemas digitales binarios.
d) Solamente los sistemas digitales binarios electrónicos.

13. ¿Qué es lo que tienen en común todos los sistemas?



Semana de la Comunidad Informática Argentino-Latinoamericana

Entre el 19 y el 23 de abril próximo, en el Sheraton Hotel de la ciudad de Buenos Aires, se realizará la **Semana de la Comunidad Informática Argentino-Latinoamericana**, integrada por dos eventos de fundamental importancia para empresas proveedoras, empresas usuarias, técnicos, científicos y especialistas en el campo de la informática.

Por un lado se realizará **Expousuaria '83 1ra. Exposición de Equipamientos, Técnicas y Servicios para la Informática**, con la participación de las más importantes empresas proveedoras, tales como **Cía. Burroughs de Máquinas Ltda., Proceda S.A., I.B.M. Argentina S.A., Hewlett Packard Argentina S.A., Texas Instruments Argentina S.A.I.C.F., Boavista Trading** de Brasil, **Mc Cormack Dodge** de Brasil, que en número de 38 ocuparán los distintos stands, presentando sus equipos y las últimas novedades en materia de computación, minicomputación, insumos y servicios.

Por otra parte se realizará el **1er. Congreso Nacional de Informática y Teleinformática "Usuaria '83 - 13 JAIIO"** que organizan en conjunto la Sociedad Argentina de Informática e Investigación Operativa, la Asociación Argentina de Usuarios de la Informática y la Federación Latinoamericana de Informática. Durante sus sesiones se desarrollarán todos los temas que son de gran interés para la comunidad informática.

La Exposición cuenta con el auspicio de la Subsecretaría de Informática de la Nación y el Congreso con los auspicios de la UNESCO y de IFIP Federación Internacional para el Procesamiento de la Información con sede en Ginebra.

Será una semana de gran trascendencia internacional, ya que llegarán al país representantes de distintas empresas de todo el mundo para asistir a ambos eventos que por primera vez se realizan en la Argentina.

La organización de **Expousuaria '83** ha sido confiada a **Inforexco S.R.L.** empresa de aquilatada trayectoria en el tema de realización de Exposiciones, Muestras y Congresos.

INFORMATICA '83

Los domingos a las 20.30 hs., por LR1 Radio "El Mundo", se irradia INFORMATICA '83, resumiendo la actividad transcurrida en la semana, con temas atinentes a la computación y a los sistemas de información, como asimismo notas y comentarios del acontecer informático. Es dirigido por el periodista Pedro Carrizo y el Lic. Carlos Tomassino.

INFORMATICA '83 es el primer programa de la radiofonía argentina orientado a esta disciplina.

Microhumor



CAMEO

Marzo de 1980

Los Angeles (ELE, API, RUTER). Dos ingenieros electrónicos desarrollaron una unidad de discos para microcomputadores. Según trascendidos formarían una compañía para su comercialización. La misma se denominaría CAMEO ELECTRONICS y tomaría como base de operaciones la localidad de Anaheim.

Enero de 1982

Nueva York. Cameo Electronics, la empresa que revolucionó el mercado de microcomputación llevando a los micros a competir con grandes computadores a precios realmente ínfimos, festejó la venta de la unidad número 1000 de su unidad de discos WD6222.

Junio de 1982

Alemania (UNSA). La compañía norteamericana Cameo Electronics instaló en esta ciudad su base de operaciones para la comercialización de sus discos rígidos en todo el ámbito de europa.

Enero de 1983

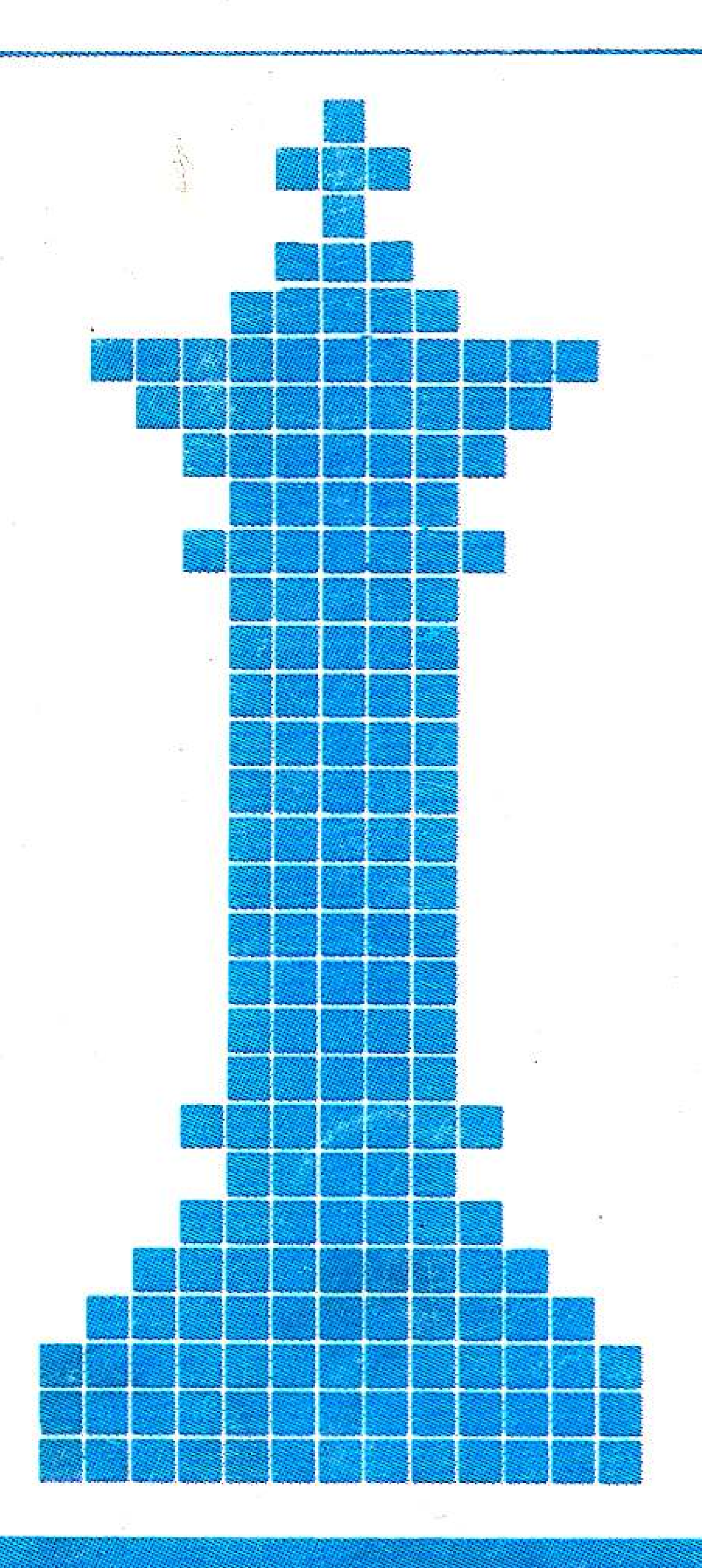
Los Angeles. Anaheim. Nuevamente la compañía Cameo Electronics revoluciona el mercado de microcomputación con la incorporación de una nueva línea de unidades de discos con capacidad de archivo de hasta 160 Mbytes para ser incorporado a cualquier microcomputador. Su precio oscilaría alrededor de los 9.000 dólares.

25 de febrero de 1983
TELEX 7941
HAB: 11:03

Confirmación. A partir del 28 de febrero del corriente año, ______________________ de Argentina representará con carácter exclusivo a Cameo Electronics Inc. en todo el territorio de Santo Domingo, Venezuela y Argentina.

Cameo Electronics Inc.
John Marshall
President

MARQUE CON UN CIRCULO EL 74 EN LA SERVICE CARD



El talento como protagonista de una sorprendente evolución tecnológica.

PROCEDA.
15 años a la vanguardia en un área imprescindible: la computación.

Una empresa en la que sus 220 especialistas altamente calificados aplican su talento e inteligencia optimizando las posibilidades de cada avance tecnológico.

Dos divisiones especializadas:

División Servicios:
Procesamiento de datos en todas sus modalidades.
Asesoramiento integral.
Diseño y programación de sistemas.

División Equipos:
Comercialización de los Computadores, Microprocesadores y Terminales Texas Instruments.
Asesoramiento Especializado.

Buenos Aires, Pueyrredón 1770 - (1119)
Tel.821-2051/0,
Córdoba, Boulev. Reconquista 178
(5000) Tel. 051 40301

